CML Microcircuits

COMMUNICATION SEMICONDUCTORS

# CMX7032
# CMX7042
# AIS Data Processor

D/7032/42_FI1.1/2 September 2006

**DATASHEET**

Provisional Issue

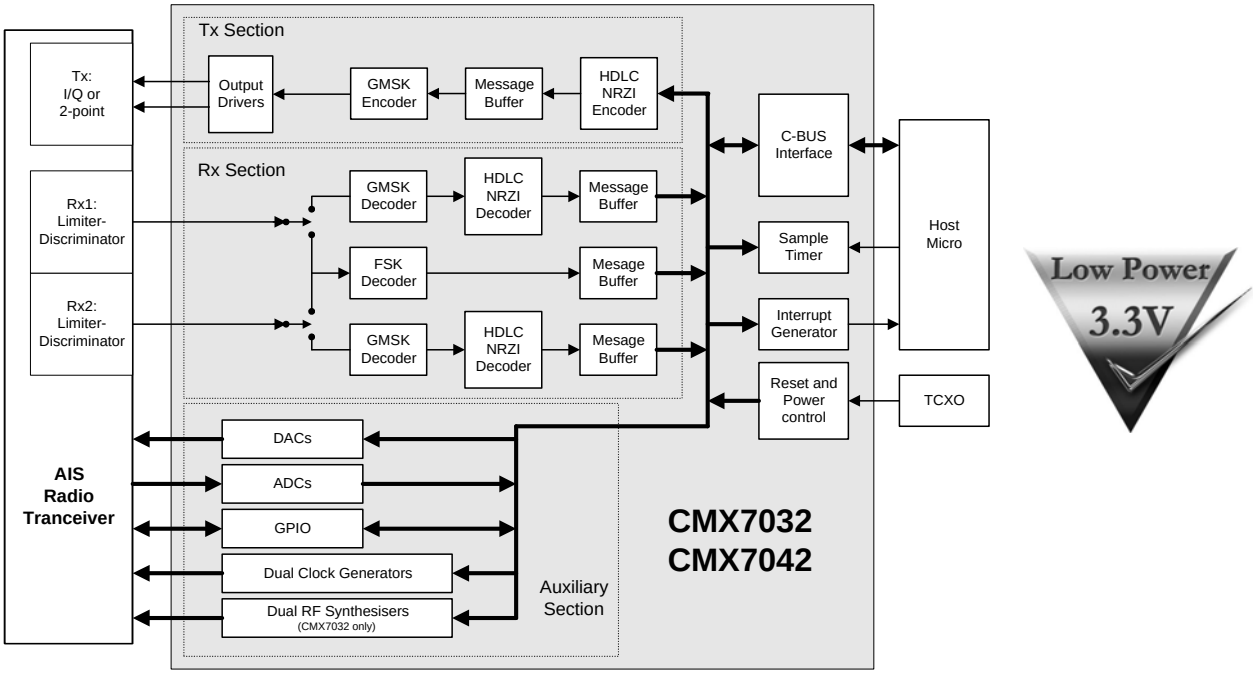
## Marine AIS data processor designed for Limiter-Discriminator based RF systems

### Features:

- Half-Duplex GMSK and FSK Modem
- AIS and DSC Data Format
- Optimum Co-channel and Adjacent-channel Performance
- Flexible Channel Configuration:
  - Two Simultaneous Rx
  - One Tx
- Supports Carrier-Sensing Channel Access (CSTDMA) Operation
- Low Profile 64 or 48 pad Leadless VQFN and 64 or 48 pin LQFP packages
- Configurable by 'Function Image'™
- Two RF synthesizers (CMX7032 only)
- Two Auxiliary System Clock Generators
- Low-Power (3.0 to 3.6V) Operation
- Limiter-Discriminator Rx Interface
- Flexible Tx Interface: I and Q or two-point modulation
- Auxiliary ADC and DAC Functions
  - 4 x (10-bit) DACs
  - 2 x (10-bit) ADCs

### Applications:

- Automatic Identification System (AIS) for Marine Safety
- AIS Aids To Navigation (AtoN)
- AIS Class B Transponders
- AIS Rx-only or Tx-only Modules







© 2006 CML Microsystems Plc

# 1 Brief Description

A highly integrated Baseband Signalling Processor IC, the CMX7032/CMX7042 fulfils the requirements of the class B marine Automatic Identification System (AIS) transponder market.

The AIS system allows ships and base stations to communicate their position and other data to each other without the need for a centralised controller. This allows vessels to "see" each other and take appropriate action to avoid collision and so improve marine safety. The system uses a GMSK 9600 baud data link in the Marine VHF radio band. The system requirements are defined in ITU-M 1371-1.

The CMX7032/CMX7042 is half duplex in operation, comprising two parallel Limiter-Discriminator Rx paths and one I and Q or two-point modulation Tx path. The Rx paths are configurable for AIS or DSC operation, the Tx for AIS only. The device performs signal modulation/demodulation with associated AIS functions, such as training sequence detection, NRZI conversion and HDLC processing (flags, bit stuffing/de-stuffing, CRC generate/check). Integrated Rx/Tx data buffers are also provided. This greatly reduces the processing requirements of the host µC. Provision of a number of auxiliary ADCs and DACs further simplifies the system hardware design, reducing the overall equipment cost and size.

The CMX7042 is identical in functionality to the CMX7032 with the exception that the two on-chip RF Synthesizers have been deleted, which enables it to be supplied in a smaller package. This document refers to both parts generically as the CMX7032, unless otherwise stated.

The CMX7032/CMX7042 devices utilise CML's proprietary *FirmASIC*® component technology. On-chip sub-systems are configured by a Function Image™: this is a data file that is uploaded during device initialisation and defines the device's function and feature set. The Function Image™ can be loaded automatically from an external EEPROM or from a host µController over the built-in C-BUS serial interface. The device's functions and features can be enhanced by subsequent Function Image™ releases, facilitating in-the-field upgrades. This document refers specifically to the features provided by Function Image™ 1.1.

This Datasheet is the first part of a two-part document comprising Datasheet and User Manual: the User Manual can be obtained by registering your interest in this product with your local CML representative.

## CONTENTS

**1 Brief Description ..... 2**

**2 Block Diagram ..... 6**

**3 Signal List ..... 7**

**4 Recommended External Components ..... 10**

**5 PCB Layout Guidelines and Power Supply Decoupling ..... 13**

**6 General Description ..... 15**
6.1 Overview ..... 15
6.2 AIS system formats ..... 16

**7 Detailed Descriptions ..... 17**
7.1 Clock Source ..... 17
7.2 Host Interface ..... 17
7.3 Function Image™ Load and Activation ..... 19
7.3.1 FI Loading from Host Controller ..... 19
7.3.2 FI Loading from EEPROM ..... 20
7.4 System Description and Tasks ..... 21
7.4.1 Signal Routing ..... 21
7.4.2 Operating Modes ..... 21
7.4.3 Modem and Data Units ..... 22
7.4.4 Timing and Synchronisation ..... 22
7.4.5 Time of Arrival Reporting ..... 22
7.4.6 Tx Timing ..... 22
7.4.7 Rx Timing ..... 23
7.4.8 CSTDMA Threshold Measurement ..... 24
7.4.9 RSSI Measurement ..... 24
7.4.10 RSSI Calibration ..... 25
7.4.11 Auxiliary ADCs ..... 25
7.4.12 Auxiliary DACs ..... 25
7.4.13 Interrupt Operation ..... 25
7.5 Operation of Tasks ..... 25
7.5.1 Tx Task Operation ..... 26
7.5.2 Rx Task Operation ..... 27
7.5.3 Registers and Buffers for Tx/Rx Tasks ..... 28
7.5.4 Write Data Registers ..... 28
7.5.5 Read Data Registers ..... 29
7.5.6 Data Tasks ..... 29
7.5.7 Modem Tasks and Codes ..... 30
7.6 Transmission format ..... 30
7.6.1 Transmit Tasks ..... 31
7.6.2 AIS Burst Mode Transmit ..... 31
7.6.3 Transmit Example ..... 32
7.6.4 AIS Raw Mode Transmit ..... 32
7.6.5 Transmitter Timing Control ..... 32
7.6.6 Modulation Formats ..... 34
7.7 Reception ..... 35
7.7.1 AIS Burst Mode Receive ..... 35
7.7.2 AIS Raw Mode Receive ..... 36
7.7.3 DSC Receive ..... 37

7.7.4 Receive Example ..... 38
7.7.5 Rx Tasks ..... 38
7.8 Configuration Tasks and Codes ..... 39
7.9 RF Synthesizer (CMX7032 only) ..... 39
7.10 System Clock Synthesizers ..... 42
7.11 Powersave ..... 43
7.12 C-BUS Register Summary ..... 44

**8 Performance Specification ..... 45**
8.1 Electrical Performance ..... 45
8.1.1 Absolute Maximum Ratings ..... 45
8.1.2 Operating Limits ..... 45
8.1.3 Operating Characteristics ..... 46
8.1.4 Parametric Performance ..... 50
8.2 C-BUS Timing ..... 51
8.3 SPI Timing ..... 52

## TABLES

Table 1 Component Values ..... 12
Table 2 BOOTEN pin states ..... 19
Table 3 Data Tasks ..... 29
Table 4 Modem Tasks ..... 30
Table 5 AIS Burst Transmit Example ..... 32
Table 6 Tx Sequence events ..... 33
Table 7 Example Tx Event Sequence Setup ..... 34
Table 8 Tx spectrum masks ..... 35
Table 9 AIS Burst Receive Example ..... 38
Table 10 Configuration Tasks ..... 39
Table 11 C-BUS registers ..... 44

## FIGURES

Figure 1 CMX7032 / CMX7042 Block Diagram ..... 6
Figure 2 CMX7032 Recommended External Components ..... 10
Figure 3 CMX7042 Recommended External Components ..... 11
Figure 4 CMX7032 Power Supply Connections and De-coupling ..... 13
Figure 5 CMX7042 Power Supply Connections and De-coupling ..... 14
Figure 6 C-BUS Transactions ..... 18
Figure 7 FI Loading from Host ..... 20
Figure 8 FI Loading from EEPROM ..... 21
Figure 9 Tx Burst Timing ..... 23
Figure 10 Rx Burst Timing ..... 24
Figure 11 Tx Task Operation ..... 26
Figure 12 Rx Task Operation ..... 27
Figure 13 Typical AIS Transmission ..... 33

Figure 14 Example RF Synthesizer Components ..... 39
Figure 15 Single RF Channel Block Diagram ..... 40
Figure 16 System Clock Generation ..... 43
Figure 17 C-BUS Timing ..... 51
Figure 18 SPI Interface Timing ..... 52
Figure 19 Mechanical outline for 64-pad VQFN package (Q1) ..... 53
Figure 20 Mechanical outline for 64-pin LQFP (leaded) package (L9) ..... 53
Figure 21 Mechanical outline for 48-pad VQFN package (Q3) ..... 54
Figure 22 Mechanical outline for 48-pin LQFP (leaded) package (L4) ..... 54

# 2 Block Diagram

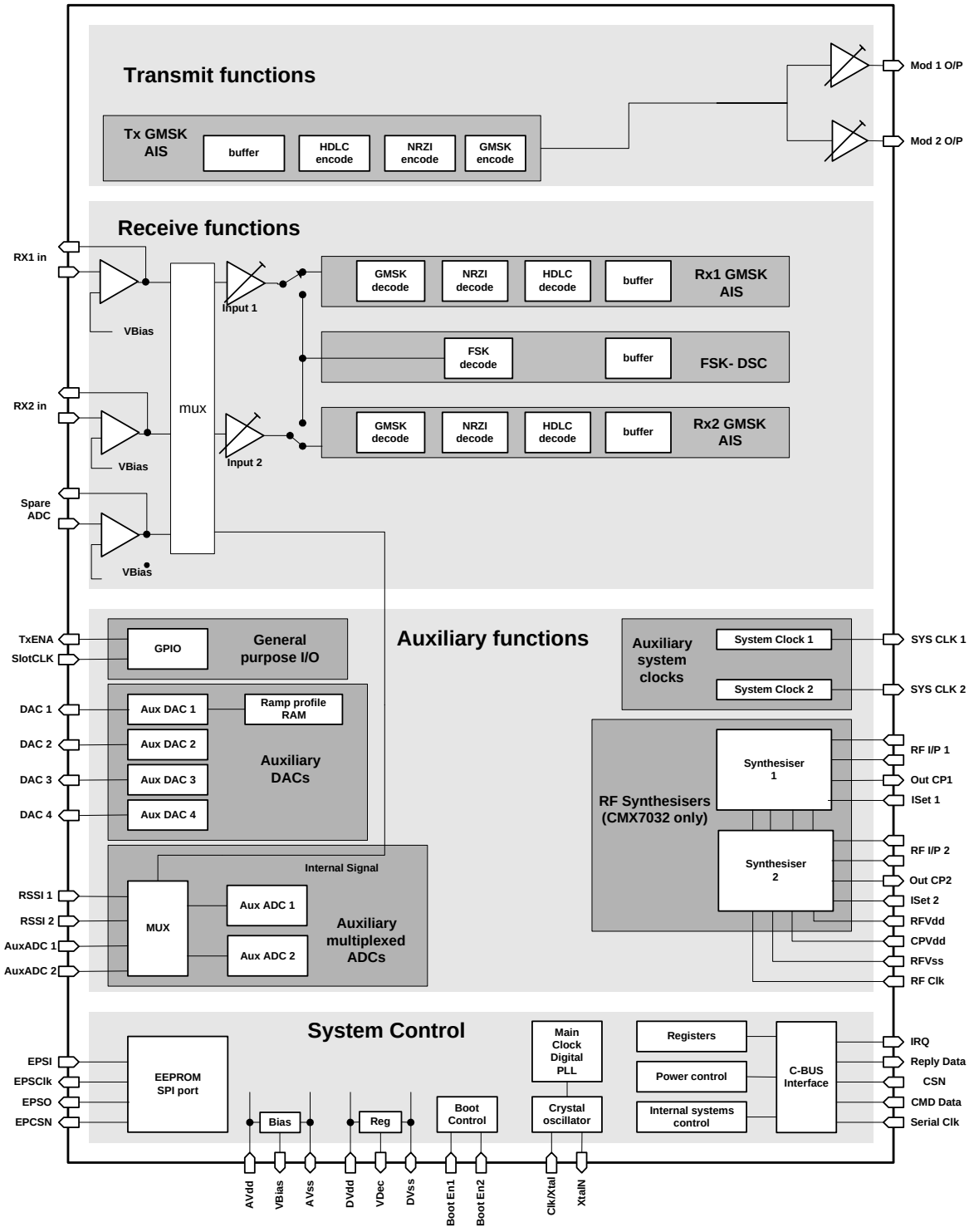



**Figure 1 CMX7032 / CMX7042 Block Diagram**

# 3 Signal List

| CMX7032 64-pin Q1 / L9 | CMX7042 48-pin Q3 / L4 | Signal Name | Type | Description |
|---|---|---|---|---|
| **1** | **8** | IRQN | OP | C-BUS: A 'wire-ORable' output for connection to the Interrupt Request input of the host. Pulled down to VSS(D) when active and is high impedance when inactive. An external pull-up resistor (R1) is required. |
| **2** | - | RF1 – | IP | RF Synthesizer #1 Negative Input. |
| **3** | - | RF1 + | IP | RF Synthesizer #1 Positive Input. |
| **4** | - | RFVss | PWR | The negative supply rail (ground) for the RF synthesizers. |
| **5** | - | CP1OUT | OP | 1st Charge Pump output. |
| **6** | - | ISET1 | IP | 1st Charge Pump Current Set input. |
| **7** | - | RFVdd | PWR | The 2.5V positive supply rail for the RF synthesizers. This should be decoupled to RFVss by a capacitor mounted close to the device pins. |
| **8** | - | RF2 – | IP | RF Synthesizer #2 Negative Input. |
| **9** | - | RF2 + | IP | RF Synthesizer #2 Positive Input. |
| **10** | - | RFVss | PWR | The negative supply rail (ground) for the 2nd RF synthesizer. |
| **11** | - | CP2OUT | OP | 2nd Charge Pump output. |
| **12** | - | ISET2 | IP | 2nd Charge Pump Current Set input. |
| **13** | - | CHARGE PUMP | PWR | The 3.3V positive supply rail for the RF charge pumps. This should be decoupled to RFVss by a capacitor mounted close to the device pins. |
| **14** | - | RF CLOCK | IP | RF Clock Input (common to both synthesizers)[1]. |
| **15** | - | - | NC | Reserved – do not connect this pin. |
| **16** | - | - | NC | Reserved – do not connect this pin. |
| **17** | - | - | NC | Reserved – do not connect this pin. |
| **18** | **9** | VDEC | PWR | Internally generated 2.5V digital supply voltage. Must be decoupled to DVss by capacitors mounted close to the device pins. No other connections allowed, except for optional connection to RFVdd. |
| **19** | **10** | SLOT CLK | IP | Slot Clock from host (37.5Hz) |
| - | **11** | - | NC | Reserved – do not connect this pin. |
| - | **12** | - | NC | Reserved – do not connect this pin. |
| **20** | **13** | SYS CLK 1 | OP | Synthesized Digital System Clock Output 1. |
| **21** | **14** | DVss | PWR | Digital Ground. |
| **22** | - | - | NC | Reserved – do not connect this pin. |
| **23** | **15** | TxEnable | OP | Enable for external Tx hardware |

[1] To minimise crosstalk, this signal should be connected to the same clock source as XTAL / CLOCK input. By default, this is connected internally at power-on, alternatively, this may be achieved by connecting the pin to the XTALN output when a 19.2MHz source is in use.

| CMX7032 64-pin Q1 / L9 | CMX7042 48-pin Q3 / L4 | Signal Name | Type | Description |
|---|---|---|---|---|
| **24** | **16** | Rx1IN | IP | Rx1 inverting input. |
| **25** | **17** | Rx1FB | OP | Rx1 input amplifier feedback. |
| **26** | **18** | Rx2IN | IP | Rx2 inverting input. |
| **27** | **19** | Rx2FB | OP | Rx2 input amplifier feedback. |
| **28** | **20** | SpareADCFB | OP | Spare ADC input amplifier feedback |
| **29** | **21** | SpareADCIN | IP | Spare ADC inverting input |
| **30** | **22** | AVss | PWR | Analog Ground. |
| **31** | **23** | MOD1 | OP | Modulator 1 output. |
| **32** | **24** | MOD2 | OP | Modulator 2 output. |
| **33** | **25** | $V_{BIAS}$ | OP | Internally generated bias voltage of about AVdd/2, except when the device is in 'Powersave' mode when VBIAS will discharge to AVss. Must be decoupled to AVss by a capacitor mounted close to the device pins. No other connections allowed. |
| **34** | **26** | - | NC | Reserved – do not connect this pin. |
| **35** | **27** | RSSI1 | IP | Analog RSSI input from Limiter / Discriminator 1. |
| **36** | **28** | RSSI2 | IP | Analog RSSI input from Limiter / Discriminator 2. |
| **37** | **29** | AUXADC1 | IP | Auxiliary ADC input 1. |
| **38** | **30** | AUXADC2 | IP | Auxiliary ADC input 2. |
| **39** | **31** | AVdd | PWR | Analog +3.3V supply rail. Levels and thresholds within the device are proportional to this voltage. This pin should be decoupled to AVss by capacitors mounted close to the device pins. |
| **40** | **32** | AUXDAC1 | OP | Auxiliary DAC output 1 / RAMDAC. |
| **41** | **33** | AUXDAC2 | OP | Auxiliary DAC output 2. |
| **42** | **34** | AVss | PWR | Analog Ground. |
| **43** | **35** | AUXDAC3 | OP | Auxiliary DAC output 3. |
| **44** | **36** | AUXDAC4 | OP | Auxiliary DAC output 4. |
| **-** | **37** | DVss | PWR | Digital Ground. |
| **45** | **38** | VDEC | PWR | Internally generated 2.5V supply voltage. Must be decoupled to DVss by capacitors mounted close to the device pins. No other connections allowed, except for the optional connection to RFVdd. |
| **46** | **39** | XTAL / CLOCK | IP | 19.2MHz input from the external clock source or 9.6MHz Xtal |
| **47** | **40** | XTALN | OP | The output of the on-chip 9.6MHz Xtal oscillator inverter. NC if 19.2MHz Clock used. |
| **48** | **41** | DVdd | PWR | Digital +3.3V supply rail. This pin should be decoupled to DVss by capacitors mounted close to the device pins. |
| **49** | **42** | COMMAND DATA | IP | C-BUS: Serial data input from the µC. |

| CMX7032 64-pin Q1 / L9 | CMX7042 48-pin Q3 / L4 | Signal Name | Type | Description |
|---|---|---|---|---|
| **50** | **43** | REPLY DATA | TS OP | C-BUS: A 3-state C-BUS serial data output to the µC. This output is high impedance when not sending data to the µC. |
| **51** | **44** | - | NC | Reserved – do not connect this pin. |
| **52** | **45** | DVss | PWR | Digital Ground. |
| **53** | - | - | NC | Reserved – do not connect this pin. |
| **54** | **46** | SERIAL CLOCK | IP | C-BUS: The C-BUS serial clock input from the µC. |
| **55** | **47** | SYS CLK 2 | OP | Synthesized Digital System Clock Output 2. |
| **56** | **48** | CSN | IP | C-BUS: The C-BUS chip select input from the µC |
| **57** | - | - | NC | Reserved – do not connect this pin. |
| **58** | **1** | EPSI | OP | EEPROM Serial Interface: SPI bus Output. |
| **59** | **2** | EPSCLK | OP | EEPROM Serial Interface: SPI bus Clock. |
| **60** | **3** | EPSO | IP+PD | EEPROM Serial Interface: SPI bus Input. |
| **61** | **4** | EPSCSN | OP | EEPROM Serial Interface: SPI bus Chip Select. |
| **62** | **5** | BOOTEN1 | IP+PD | Used in conjunction with BOOTEN2 to determine the operation of the bootstrap program. |
| **63** | **6** | BOOTEN2 | IP+PD | Used in conjunction with BOOTEN1 to determine the operation of the bootstrap program. |
| **64** | **7** | DVss | PWR | Digital Ground. |

**Notes:** Exposed metal pad on the bottom of a VQFN package should not be connected.
IP = Input (+PU/PD = internal pullup/pulldown resistor)
OP = Output
TS OP = 3-state Output
PWR = Power Supply Connection
NC = No Connection
Functions with no associated pin number are not available in the CMX7042.

# 4 Recommended External Components

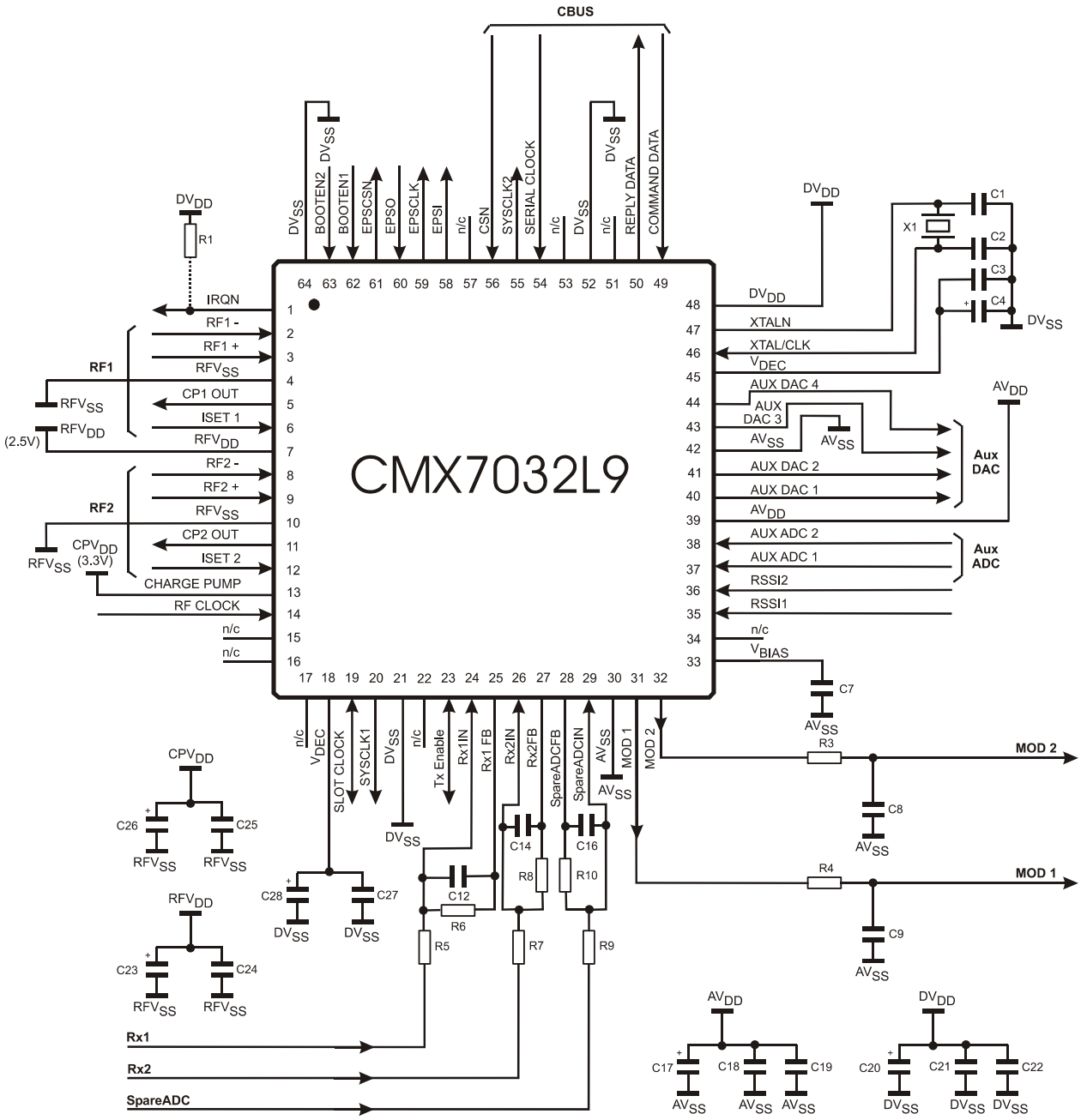



**Figure 2 CMX7032 Recommended External Components**

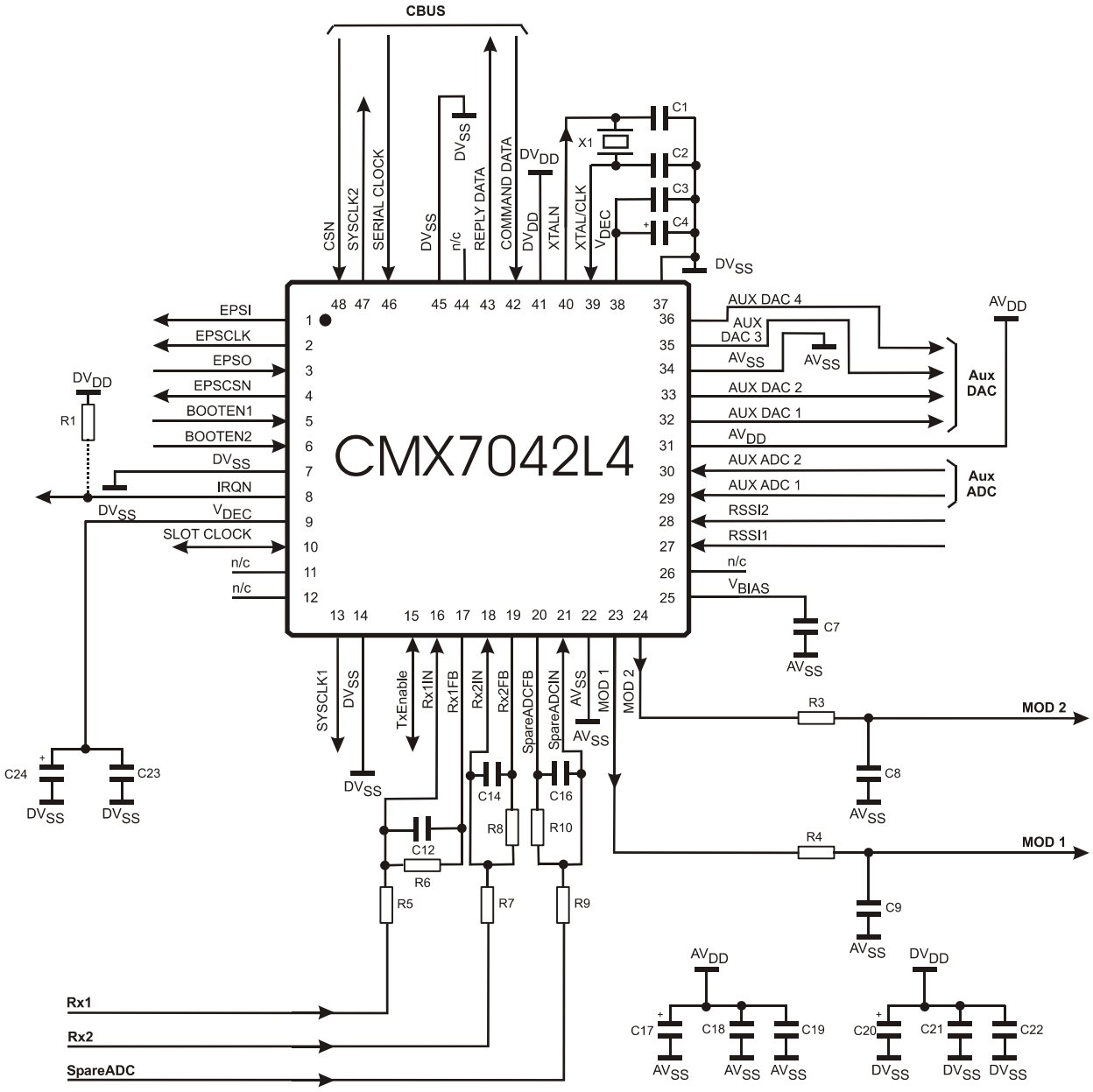



**Figure 3 CMX7042 Recommended External Components**

**Table 1 Component Values**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R1 | 100kΩ | C1 | 18pF | C16 | 100pF | C21 | 10nF |
| R3 | 100kΩ | C2 | 18pF | C17 | 10µF | C22 | 10nF |
| R4 | 100kΩ | C3 | 10nF | C18 | 10nF | C23 | 10µF |
| R5 | See note 2 | C4 | 10µF | C19 | 10nF | C24 | 10nF |
| R6 | 100kΩ | C7 | 100nF | C20 | 10µF | C25 | 10nF |
| R7 | See note 3 | C8 | 100pF | | | C26 | 10µF |
| R8 | 100kΩ | C9 | 100pF | | | X1 | 9.6MHz |
| R9 | See note 4 | C12 | 100pF | | | | See note 1 |
| R10 | 100kΩ | C14 | 100pF | | | | |

Resistors ±5%, capacitors and inductors ±20% unless otherwise stated.

Notes:

1. X1 can be a 9.6MHz crystal or a 19.2MHz external clock generator. The tracks between the crystal and the device pins should be as short as possible to achieve maximum stability and best start up performance.
2. R5 should be selected to provide the desired dc gain (assuming C11 is not present) of the input, as follows:

$$|GAIN_{RxIN1}| = 100k\Omega / R5$$

   The gain should be such that the resultant output at the RxIN1FB pin is within the input signal range specified in 8.1.3.
3. R7 should be selected to provide the desired dc gain (assuming C13 is not present) of the input as follows:

$$|GAIN_{RxIN2}| = 100k\Omega / R7$$

   The gain should be such that the resultant output at the RxIN2FB pin is within the input signal range specified in 8.1.3.
4. R9 should be selected to provide the desired dc gain (assuming C15 is not present) of the input as follows:

$$|GAIN_{SpareADC}| = 100k\Omega / R9$$

   The gain should be such that the resultant output at the RxIN2FB pin is within the input signal range specified in 8.1.3.
5. Care should be taken in connecting the output of the Limiter-Discriminator device to the RxIN pins of the CMX7032. The format of the GMSK signal requires that the frequency response of the input circuits extends to below 10Hz, however the variations in the incoming AIS signals from many different stations require that the input must rapidly follow the changes in DC and signal levels without de-grading the signal seen at the RxIN pin.

© 2006 CML Microsystems Plc

# 5 PCB Layout Guidelines and Power Supply Decoupling

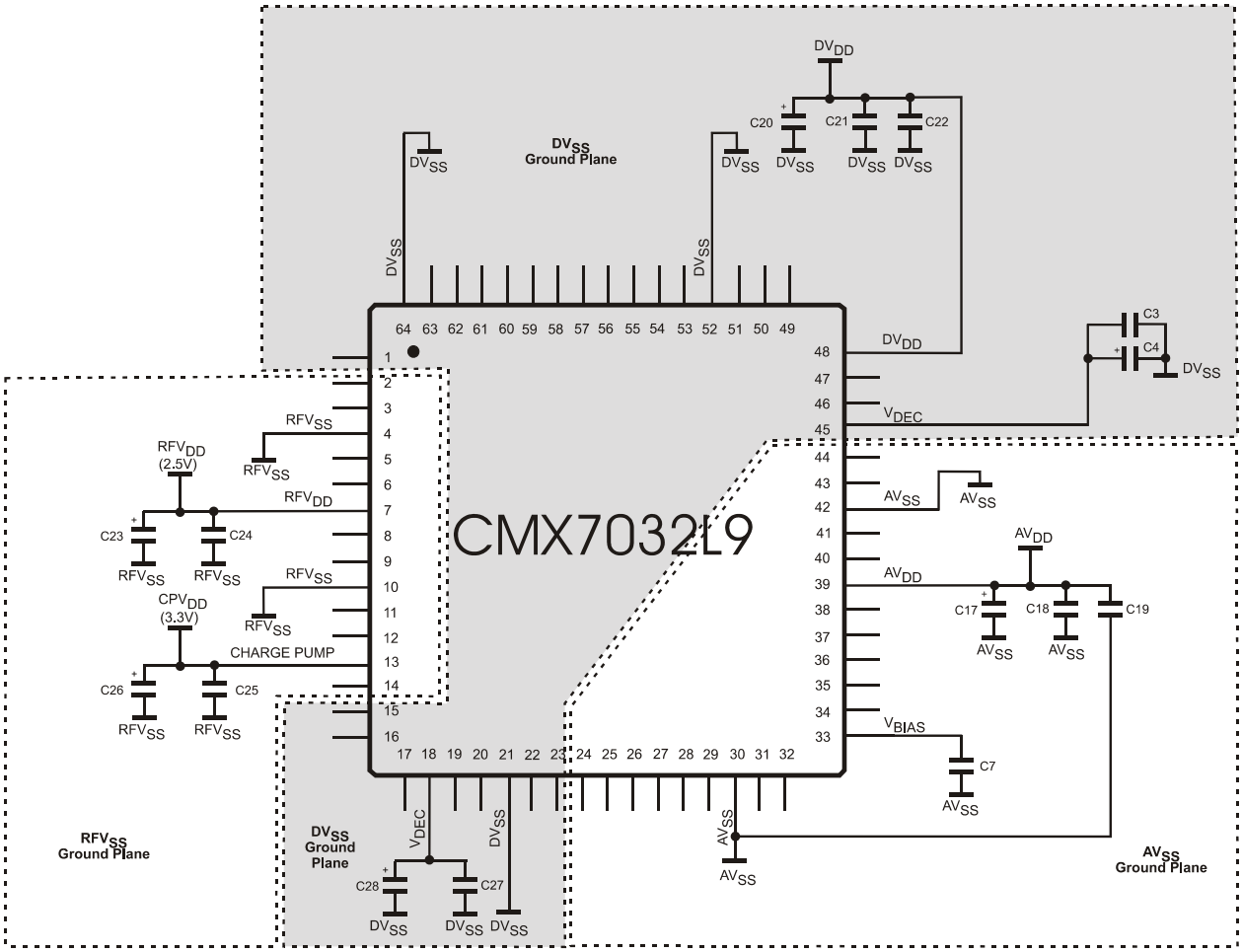



**Figure 4 CMX7032 Power Supply Connections and De-coupling**

Component Values as per Table 1.

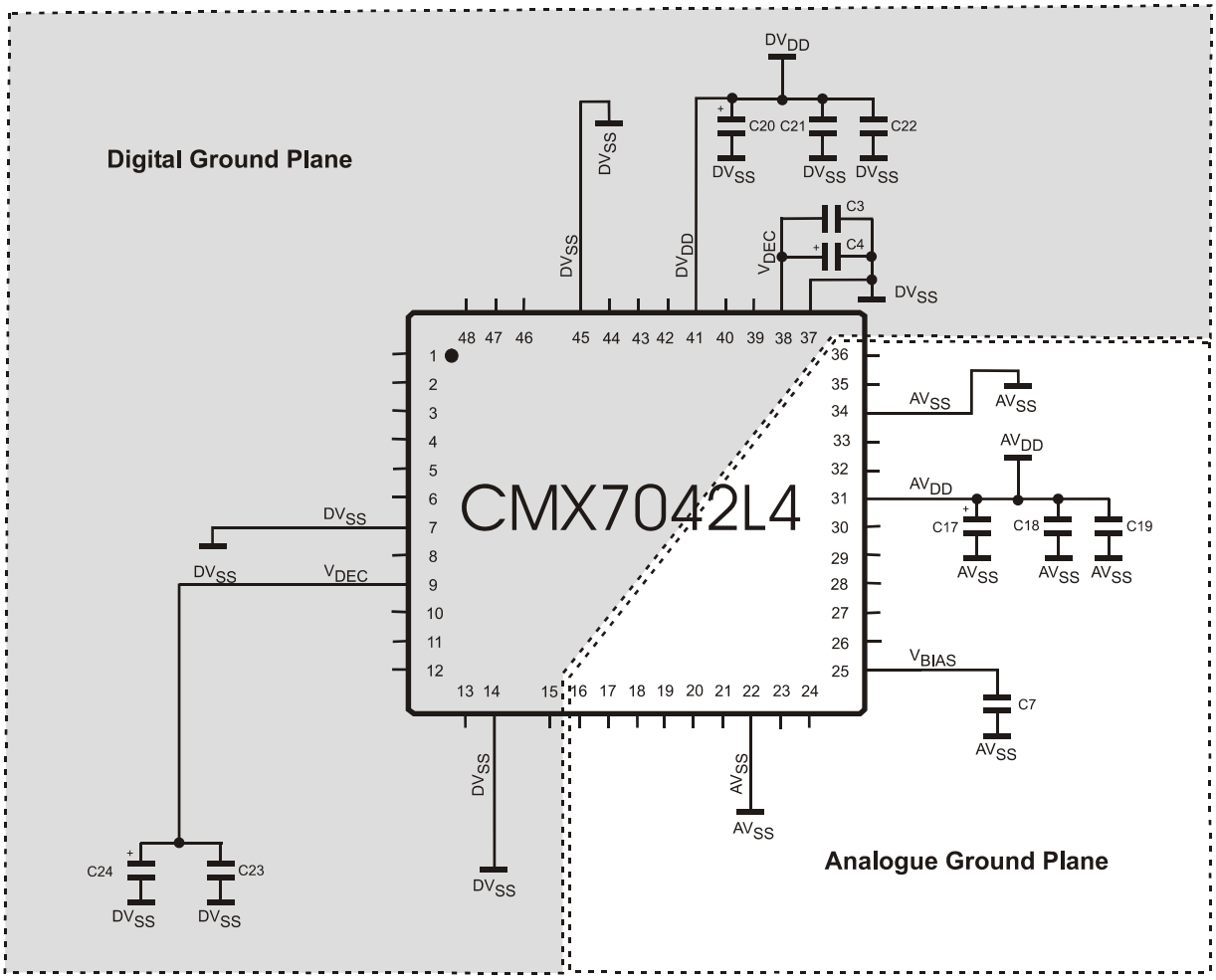



**Figure 5 CMX7042 Power Supply Connections and De-coupling**

Component Values as per Table 1.

Notes:

1. It is important to protect the analog pins from extraneous inband noise and to minimise the impedance between the CMX7032 and the supply and bias de-coupling capacitors. The supply decoupling capacitors should be as close as possible to the CMX7032. It is therefore recommended that the printed circuit board is laid out with separate ground planes for the $AV_{SS}$, $RFV_{SS}$ and $DV_{SS}$ supplies in the area of the CMX7032, with provision to make links between them, close to the CMX7032. Use of a multi-layer printed circuit board will facilitate the provision of ground planes on separate layers.

2. It is recommended that no connection be made to the central metal pad on Q1,Q3 (VQFN) packages.

3. $V_{BIAS}$ is used as an internal reference for detecting and generating the various analog signals. It must be carefully decoupled to ensure its integrity so, apart from the decoupling capacitor shown, no other loads should be connected. If $V_{BIAS}$ needs to be used to set the discriminator mid-point reference, it must be buffered with an external high input impedance buffer.

4. The 2.5V $V_{DEC}$ output can be used to supply the 2.5V $RFV_{DD}$, to remove the need for an external 2.5V regulated supply. $V_{DEC}$ can be directly connected to $RFV_{DD}$.

# 6 General Description

## 6.1 Overview

**Tx Modem Functions**

- AIS 25kHz channel (GMSK, 9600bps, 2.4kHz deviation, BT=0.4)
- AIS Burst mode with full AIS frame formatting (HDLC-type)
    - Bit stuffing
    - NRZI coding
    - Training sequence and start/stop flag insertion
    - CRC generation
- AIS Raw mode (for greater flexibility)
    - Supports arbitrary data streams for user-defined protocols
- Full support for the AIS Class B carrier-sensing channel access scheme (CS-TDMA)
- Full support for the AIS Class B Self-organising TDMA access scheme
- 160 byte (equivalent to 5 AIS slots) Tx data buffer
- Flexible Tx Interface
    - Two-point modulation outputs, with independent gain and polarity controls
    - I and Q modulation outputs, for use with an I/Q upconverter

**Rx Modem Functions**

- Configurable modulation format:
    - AIS 25kHz channel (GMSK, 9600bps, 2.4kHz deviation, BT=0.4)
    - DSC (de-emphasised FSK, 1200bps)
- Simultaneous reception of two AIS channels, or one AIS and one DSC channel
- AIS Burst mode with full AIS frame formatting (HDLC-type)
    - Frame sync recognition
    - Bit de-stuffing
    - NRZI decoding
    - Training sequence and start/stop flag detection
    - CRC checking
- AIS Raw mode (for greater flexibility)
    - Supports arbitrary data streams for user-defined protocols
- DSC reception supported in raw mode
- Four 160 byte Rx data buffers can automatically store up to four 5-slot AIS bursts (2 per Rx channel)
- Rx signal input gain and polarity controls
- Time-of-arrival reporting assists with timekeeping in the absence of GPS

**100 - 600 MHz RF Synthesizers (CMX7032 only)**

- Two Integer-N synthesizers
- Flexible design minimizes reference spurs for low phase noise results
- Charge pump
    - High/low soft selectable current setting to speed large frequency channel changes
    - Nominal current user defined by external resistor value
- Lock detect

**Analog I/O Functions**

- Auxiliary ADC system
  - Two 10-bit successive approximation ADCs with integrated sample and hold.
  - One AuxADC is dedicated to RSSI measurement in a user defined window, the other is available for general use.
- Ramping auxiliary DAC
  - DAC sequences through a user-configured sequence of DAC output values to develop a specific rising/falling DAC output signal. This is useful for ramping an RF PA, and can be configured to happen automatically at the start and end of a burst.
  - Can operate as a general purpose DAC, if desired.
- Three general purpose auxiliary DACs

**System Functions**

- All internal subsystems are controlled via a single serial host interface to reduce host µC pin count and simplify external host driver complexity.
- Transaction oriented command/response logical host interface executes tasks supporting normal operation, device configuration, and functions to assist manufacturing calibration trimming of external circuits.
- Internal system clock derived from RF synthesizer reference oscillator and eliminates the need for additional XTAL or baseband clock oscillator.
- User clock synthesizers generate two clocks for external use and eliminates an external clock synthesizer when needed to support peripheral devices.
- User selected method for loading a Function Image™ - either from the host µC via C-BUS or from an external serial EEPROM.
- Integrated 2.5V regulator can develop 2.5V from required 3.3V supply.
- Powersave facilities minimize total system power.

### 6.2 AIS system formats

The AIS system uses two basic channel access mechanisms – Self Organising Time Division Multiple Access (SOTDMA) and Carrier-Sensing Time Division Multiple Access (CSTDMA). The CMX7032 is compatible with both systems and offers additional features which simplify the implementation of CSTDMA. The SOTDMA system is detailed in ITU-M 1371-1 and IEC 61993-2 while the CSTDMA is detailed in IEC 62287.

The CSTDMA system is used in the implementation of the Class B-CS AIS. This requires the Receiver to monitor the first part of a slot for an existing AIS transmission from another station before deciding to use the slot for its own transmission or aborting and selecting a different slot.

The SOTDMA system is used in the Class A and Base Station AIS as well as the Class B-SOTDMA AIS standard. The relevant International standards are:

| | |
|---|---|
| [0] | ITU-M 1371-1 |
| [1] | IEC 61993-2 Class A |
| [2] | IEC 62287-1 Class B CSTDMA |
| [3] | IEC 62287-2 Class B SOTDMA |
| [4] | IEC 62320-1 Base Station |
| [5] | IEC 62320-2 Aids to Navigation |

# 7 Detailed Descriptions

## 7.1 Clock Source

The CMX7032 can be used with either a 9.6MHz xtal or a 19.2MHz oscillator. The RFCLOCK should also be derived from this source to avoid the generation of unwanted spurious signals.

## 7.2 Host Interface

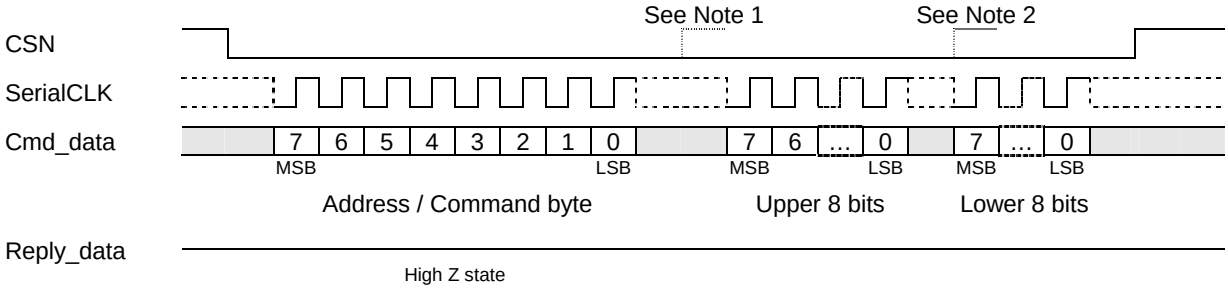
This section provides a general description of the C-BUS serial interface protocol used to transfer data, control and status information between the CMX7032 and its host.

C-BUS is a serial interface, similar to SPI, that uses a simple transaction-oriented command/response protocol with addressing to access specific registers within the CMX7032. Each C-BUS transaction consists of a single Register Address/Command byte (A/C byte) sent from the µC which may be followed by one or more data byte(s) sent from the µC to be written into one of the CMX7032's Write Only Registers, or one or more data byte(s) read out from one of the CMX7032's Read Only Registers, as illustrated in Figure 6.

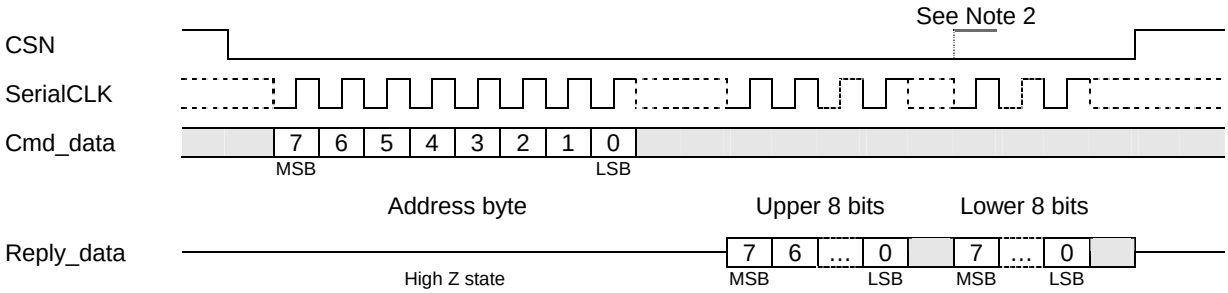
Data sent from the µC on the Command Data line is clocked into the CMX7032 on the rising edge of the Serial Clock input. Reply Data sent from the CMX7032 to the µC is valid when the Serial Clock is high. The CSN line must be held low during a data transfer and kept high between transfers. The C-BUS interface is compatible with most common µC serial interfaces and may also be easily implemented with general purpose µC I/O pins controlled by a simple software routine.

The number of data bytes following an A/C byte is dependent on the value of the A/C byte. The most significant bit of the address or data is sent first. For detailed timings see section 8.2.

**C-BUS Write:**




**C-BUS Read:**




Data value unimportant  Repeated cycles  Either logic level valid

**Figure 6 C-BUS Transactions**

Notes:

1. For Command byte transfers only the first 8 bits are transferred.
2. For single byte data transfers only the first 8 bits of the data are transferred.
3. The Cmd_data and Reply_data lines are never active at the same time. The Address byte determines the data direction for each C-BUS transfer.
4. The SerialCLK input can be high or low at the start and end of each C-BUS transaction.
5. The gaps shown between each byte on the Cmd_data and Reply_data lines in the above diagram are optional, the host may insert gaps or concatenate the data as required.

### 7.3 Function Image™ Load and Activation

The Function Image™ (FI), which defines the operational capabilities of the device, may be obtained from the CML Technical Portal, following product registration. This is in the form of a 'C' header file which can be included into the host controller software or programmed into an external EEPROM. The maximum possible size of Function Image™ is 46 kbytes, although a typical FI will be less than this. Note that the BOOTEN pins are only read at power-on or following a C-BUS General Reset and must remain stable throughout the FI loading process. Once the FI load has completed, the BOOTEN pins are ignored by the CMX7032 until the next power-up or C-BUS General Reset.

The BOOTEN pins are both fitted with internal 100k (approx.) pull down resistors.
For C-BUS load operation, both pins should be pulled high by connecting them to $V_{dd}$ either directly or via a 4k7 resistor (see Table 2).
For EEPROM load, only BOOTEN1 needs to be pulled high in a similar manner, however, if it is required to program the EEPROM in-situ from the host, either a jumper to $V_{dd}$ or a link to a host I/O pin should be provided to pull BOOTEN2 high when required (see Table 2).

Once the FI has been loaded, the CMX7032 performs these actions:-

(1) the product identification code ($7032 or $7042) is reported in C-BUS register $C5
(2) the FI version code is reported in C-BUS register $C9
(3) the two 32-bit FI checksums are reported in C-BUS register pairs $A9, $AA and $B8, $B9
(4) the device waits for the host to load the 32-bit Device Activation Code to C-BUS register $C8
(5) once activated, the device initialises fully, enters idle mode and becomes ready for use.

The checksums can be verified against the published values to ensure that the FI has loaded correctly. Once the FI has been activated, the checksum, product identification and version code registers are cleared and these values are no longer available. If an invalid activation code is loaded, the device will report the value $DEAD in register $A9 and become unresponsive to all further host commands (including General Reset).

Both the Device Activation Code and the checksum values are available from the CML website.

**Table 2 BOOTEN pin states**

| | BOOTEN2 | BOOTEN1 |
|---|---|---|
| C-BUS Host load | 1 | 1 |
| reserved | 1 | 0 |
| EEPROM load | 0 | 1 |
| No FI load | 0 | 0 |

Note: In the rare event that a General Reset needs to be issued without the requirement to re-load the FI, the BOOTEN pins must both be cleared to '0' before issuing the Reset command. The Checksum values will be reported and the Device Activation code will need to be sent in a similar manner as that shown in Figure 8. There will not be any FI loading delay. This assumes that a valid FI has been previously loaded and that Vdd has been maintained throughout the reset to preserve the data.

#### 7.3.1 FI Loading from Host Controller

The FI can be included into the host controller software build and downloaded into the CMX7032 at power-up over the C-BUS interface. The BOOTEN pins must be set to the C-BUS load configuration, the CMX7032 powered up and placed into Program Mode, the data can then be sent directly over the C-BUS to the CMX7032.

Each time the device is powered up its Function Image™ must first be loaded and then activated. These two steps assign internal device resources and determine all device features. The device does not operate until the function image is loaded and activated.

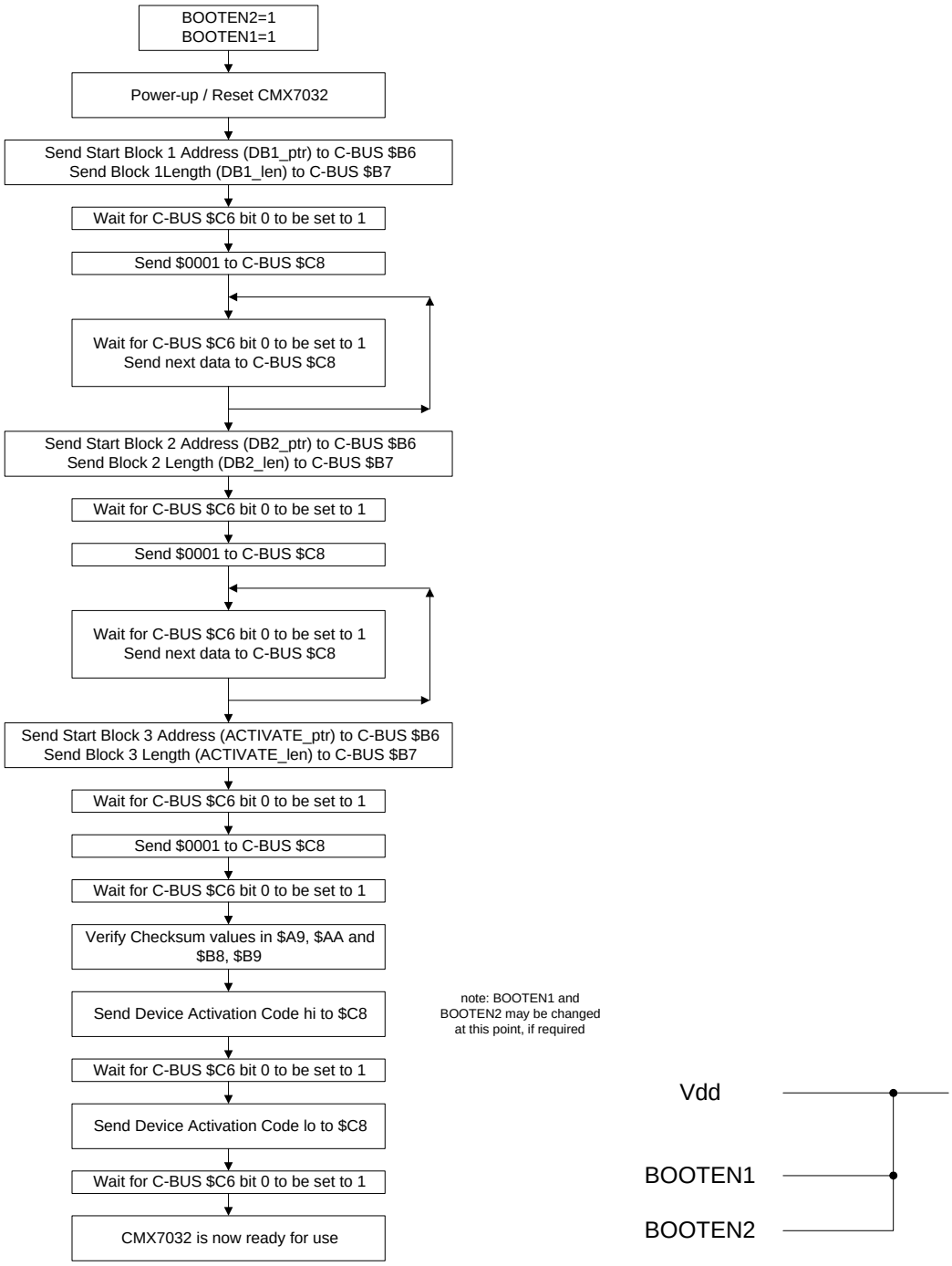

BOOTEN2=1
BOOTEN1=1
Power-up / Reset CMX7032
Send Start Block 1 Address (DB1_ptr) to C-BUS $B6
Send Block 1Length (DB1_len) to C-BUS $B7
Wait for C-BUS $C6 bit 0 to be set to 1
Send $0001 to C-BUS $C8
Wait for C-BUS $C6 bit 0 to be set to 1
Send next data to C-BUS $C8
Send Start Block 2 Address (DB2_ptr) to C-BUS $B6
Send Block 2 Length (DB2_len) to C-BUS $B7
Wait for C-BUS $C6 bit 0 to be set to 1
Send $0001 to C-BUS $C8
Wait for C-BUS $C6 bit 0 to be set to 1
Send next data to C-BUS $C8
Send Start Block 3 Address (ACTIVATE_ptr) to C-BUS $B6
Send Block 3 Length (ACTIVATE_len) to C-BUS $B7
Wait for C-BUS $C6 bit 0 to be set to 1
Send $0001 to C-BUS $C8
Wait for C-BUS $C6 bit 0 to be set to 1
Verify Checksum values in $A9, $AA and $B8, $B9
Send Device Activation Code hi to $C8
Wait for C-BUS $C6 bit 0 to be set to 1
Send Device Activation Code lo to $C8
Wait for C-BUS $C6 bit 0 to be set to 1
CMX7032 is now ready for use
note: BOOTEN1 and BOOTEN2 may be changed at this point, if required
Vdd
BOOTEN1
BOOTEN2

**Figure 7 FI Loading from Host**

The download time is limited by the clock frequency of the C-BUS, with a 5MHz SCLK, it should take less than 500ms to complete.

### 7.3.2 FI Loading from EEPROM

The FI must be converted into a format for the EEPROM programmer (normally Intel Hex) and loaded into the EEPROM either by the host or an external programmer. The CMX7032 needs to have the BOOTEN pins set to EEPROM load, and then on power-on, or following a C-BUS General Reset, the CMX7032 will automatically load the data from the EEPROM without intervention from the host controller.

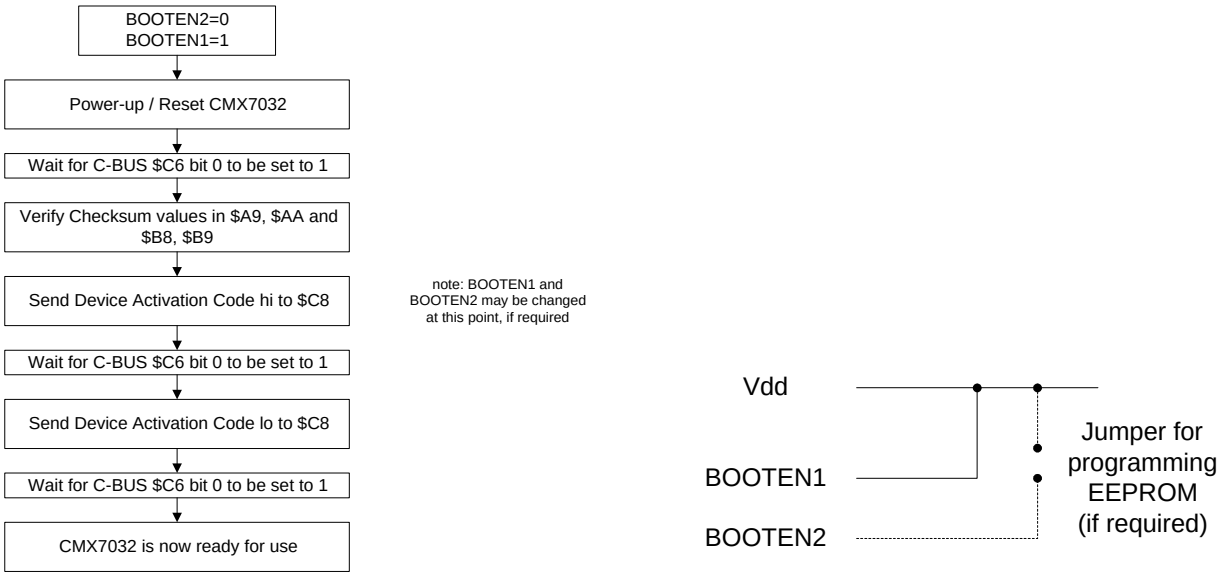

BOOTEN2=0
BOOTEN1=1
Power-up / Reset CMX7032
Wait for C-BUS $C6 bit 0 to be set to 1
Verify Checksum values in $A9, $AA and $B8, $B9
Send Device Activation Code hi to $C8
Wait for C-BUS $C6 bit 0 to be set to 1
Send Device Activation Code lo to $C8
Wait for C-BUS $C6 bit 0 to be set to 1
CMX7032 is now ready for use
note: BOOTEN1 and BOOTEN2 may be changed at this point, if required
Vdd
BOOTEN1
BOOTEN2
Jumper for programming EEPROM (if required)

**Figure 8 FI Loading from EEPROM**

The CMX7032 has been designed to function with Atmel AT25HP512 EEPROM devices, however other manufacturers parts may also be suitable. The time taken to load the FI is dependant on the clock / Xtal source frequency, but should be less than 500ms.

### 7.4 System Description and Tasks

This section describes the operation of main sections of the CMX7032 and the task-oriented logical interface provided to the external host device.

#### 7.4.1 Signal Routing

The CMX7032 provides processing capability for two simultaneous receive channels (either two AIS or one AIS and one DSC) or one Tx channel (AIS).

The inputs to the two receiver channels are nominally allocated to the Rx1IN and Rx2IN pins of the device, however they can be re-allocated by the user by use of the Input Signal Routing C-BUS register, $B1. The SpareADCIN pin can also be used as an input to the receivers if required. By default the device will route Rx1IN to Rx channel 1 and Rx2IN to Rx channel 2, which leaves the SpareADCIN signal available to the AuxADC. All of these inputs are configured around an inverting op-amp stage to facilitate gain and filtering adjustments. In addition, the Rx channel inputs are equipped with programmable gain stages for further level adjustments as required.

The Tx Modulation output signals may be configured to be suitable for two-point modulation circuits or alternatively an I / Q upconverter. Signal levels on both output pins, MOD1 and MOD2 can be set to within 0.2dB using a Configuration Mode task.

#### 7.4.2 Operating Modes

The CMX7032 operates in either:

- Configuration mode.
- Normal mode.

Configuration mode is used to set up various operating parameters of the CMX7032 subsystems e.g. Transmit format, timing parameters etc. following a power-up or reset. The modem section is disabled when the device is in Configuration mode. Configuration mode uses dedicated tasks that are not valid whilst in Normal mode.

Normal mode is used when actively running the CMX7032 modem and other subsystems. Normal mode uses dedicated tasks that are not valid whilst in Configuration mode.

"Enter config mode" (ECM) is a Normal mode task that switches the device from Normal to Configuration mode. "Exit config mode" (EXIT_CONFIG) is a Configuration mode task that switches the device from Configuration to Normal mode.

### 7.4.3 Modem and Data Units

The CMX7032 is logically divided into two main units which can accept and perform tasks separately:

- Modem Unit
- Data Unit

The Modem Unit is primarily responsible for processing Tx data from the internal Tx data buffer and presenting it on the MOD1 and MOD2 pins, and processing the Rx input signals to recover the Rx data they represent and storing that data in the internal Rx data buffers.

The Data Unit is primarily responsible for transferring data between the internal data buffers or subsystems and the C-BUS registers, from where they can be accessed by the host μC.

When the device is in Normal mode, the Command Register, $C8, is a 16 bit C-BUS write register that contains task fields for both Data and Modem units. A task is invoked by writing its code into the Data Task or Modem Task fields. A single C-BUS write transaction will change all Command Register fields. Often, the host will only want to issue either a Data or Modem task, in which case it should ensure that the other task field is set to all zeroes, corresponding to a null / idle task. Sometimes it is useful to issue Data and Modem tasks simultaneously, in which case, the Data task will always be completed before the Modem task is started.

Certain internal subsystems can be directly accessed and controlled via C-BUS transactions, without issuing a specific task / command.

### 7.4.4 Timing and Synchronisation

An AIS transponder must keep track of both the current AIS slot number and the position within the slot. This is principally to ensure that transmissions occur at the right time, and that hardware is correctly switched between Rx and Tx, but is also useful for scheduling when to take RSSI measurements, when receivers can be powered down and when they should wake up again.

In an AIS Class A transponder, the clock is synchronised with a 1Hz tick from a GPS unit. In a Class B transponder the clock may be synchronised to the GPS tick, or may be synchronised to the reception time of AIS bursts from a Class A transponder or Base Station. Note that the latter scheme requires management by the host μC, which must determine which received bursts are qualified to be used as a timing reference. Whichever of these methods is used by the transponder, the CMX7032 requires a SlotCLK input from the host μC. This should be a pulse at least 50us long, whose rising edge is aligned to the AIS SlotCLK. An edge is required at the start of every AIS slot, hence the frequency of this signal is 37.5Hz.

The CMX7032 has several features to assist the host μC with timing, which are detailed below. All of these features are based on the SlotCLK signal, provided by the host to the CMX7032's SlotCLK pin. All timings are defined as a number of 24kHz "ticks" referenced to the rising edge of the SlotCLK signal.

### 7.4.5 Time of Arrival Reporting

When the CMX7032 has received a burst as the result of a RXB1/2 (receive burst) task, the time of arrival is presented as one of the first four words of the Data Block. This will give the time, measured in 24kHz ticks, between the rising edge of the last SlotCLK and the detection of the last bit of the start flag of the burst ($T_{sync}$ in ITU-M 1371-1).

### 7.4.6 Tx Timing

The CMX7032 can be configured to perform a sequence of events when a TXB or TDBS task (transmit burst) is issued. The events are start and end of modulation, ramping the RAMDAC up and down, asserting and releasing a digital output pin (intended as a Tx Enable) and CSTDMA sensing. Each of

these can be configured to happen with specified delays from the rising edge of the SlotCLK. The timings are set up with the config mode task Tx_Sequence. See User Manual section 11.15.7 for details.

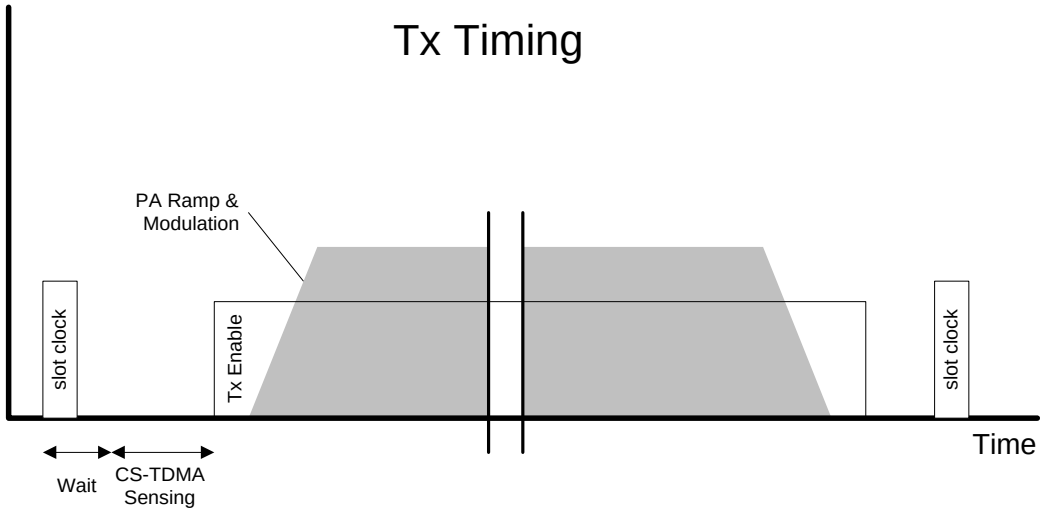



**Figure 9 Tx Burst Timing**

If CSTDMA sensing has been enabled then the CMX7032 will measure the carrier level (RSSI) on the selected channel during the window specified in the transmit timing table. If the user-specified threshold is exceeded then the subsequent events (RAMDAC, Tx Enable and modulation) will be cancelled. The threshold is set using the CS_Threshold register, $C2. See User Manual section 11.12 for details.

### 7.4.7 Rx Timing

The CMX7032 has a power-saving mode, where it spends most of its time ‘asleep’, but ‘wakes up’ at the start of each AIS slot and receives for long enough to determine whether a burst is present or not. If no burst is received the CMX7032 will make a pair of RSSI measurements and then go back to sleep. This power saving operation can be turned on and off using the Sleep Mode En bit in the Mode register, $C1. The CMX7032 will still respond to C-BUS accesses when ‘asleep’ but the latency between a task being issued and completed may be longer than normal.

The times to perform RSSI measurements, go to sleep, and wake up are configured by the host, and all times are referenced to the SlotCLK signal. The host should ensure that these timings result in the CMX7032 being awake in time to sample the next SlotCLK signal. The timings are set up using the Sleep_Timing Config task. See User Manual section 11.15.8 for details.

If Sleep mode is not enabled, then the demodulator will run continuously and update the Rx Data buffer whenever valid data is received.

© 2006 CML Microsystems Plc

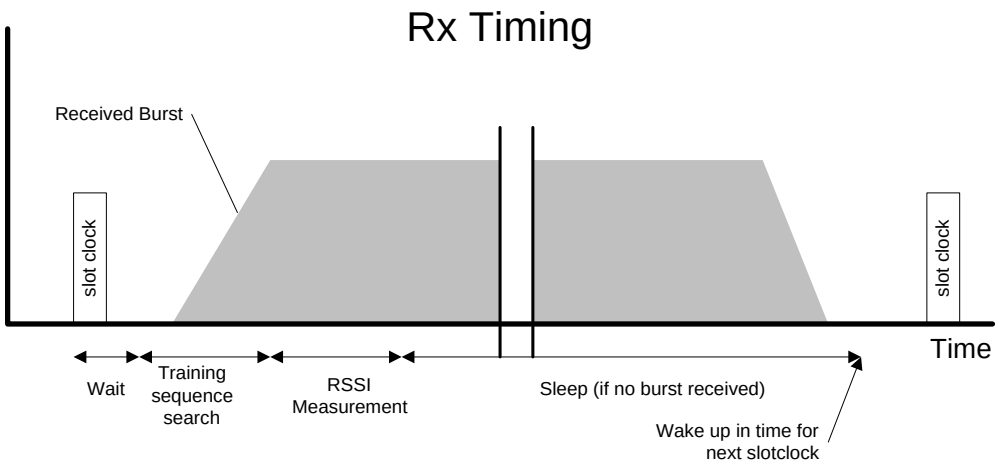



**Figure 10 Rx Burst Timing**

### 7.4.8 CSTDMA Threshold Measurement

The CMX7032 provides support for the CSTDMA mode of operation by measuring the signal level at the start of a slot period and aborting the transmission if a level over a user defined threshold is found. This fact is reported to the host in Status2 C-BUS register $C5.

The CSTDMA threshold measurement window is setup by the Tx_Sequence Config task using timing values based on the number of 24kHz "tick" increments from the rising edge of the SlotCLK pulse. During this period (if Tx_Status and CS-ENA are both "active") the CMX7032 will measure the RSSI level of the appropriate channel (set in C-BUS Mode register $C1):

1. sample RSSI at 48k samples/sec
2. scale values to 0 -128
3. apply the look-up table values
4. apply the CSTDMA_gain value
5. accumulate values over the defined period
6. compare the value with the value in the C-BUS CS-Threshold register $C2 and abort the transmission if it is exceeded.

Note that the same lookup table is used for both CSTDMA Threshold measurement as the RSSI measurement.

### 7.4.9 RSSI Measurement

RSSI can not be determined directly from the baseband signal output of a limiter-discriminator device so the CMX7032 has two dedicated inputs (one for each Rx channel) for separate external analogue RSSI signals. Suitable outputs are available on many limiter-discriminator ICs, e.g. Philips SA605. The ADCs sample the signals during a user-defined window and apply an averaging algorithm and present the results in C-BUS registers $BA and $BB.

The RSSI measurement windows are set up using the RSSI_Window config task using timing values based on the number of 24kHz "tick" increments from the rising edge of the SlotCLK plulse. During this period (if Tx_Status is not "active") the device will:

1. sample RSSI at 48k samples/sec
2. scale values to 0 -128
3. apply the look-up table values
4. apply the RSSI_gain value
5. accumulate values over the defined period
6. output values to C-BUS registers

Note that the RSSI measurements on each RF channel run consecutively and that the C-BUS registers are updated at the end of the RSSI_2 measurement window.

### 7.4.10 RSSI Calibration

The RSSI signal should be approximately logarithmically scaled (i.e. a nearly linear relationship between voltage and signal strength in dBs). In order to reduce the impact of noise, the CMX7032 averages over several samples, but to give a meaningful average, the samples must first be anti-logged. In order to correct any non-linearities in the RSSI response, and to set an offset for the anti-logging the host must supply calibration data.

The calibration data is a 128-entry table. The entries correspond to equally-spaced voltages from 0V to 3.3V applied to the RSSI inputs. To set up the table the host uses the Setup_RSSI_Calibration config task. See section 7.8 for details. The default values are shown in User Manual section 11.15.3.

### 7.4.11 Auxiliary ADCs

The first Auxiliary ADC is dedicated to RSSI measurements at times specified by the host uC, see Section 7.4.8 and User Manual 11.15.4. The second Auxiliary ADC is available for user functions. The ADC runs continuously, the input selected by the AuxADC Input Select bits in the C-BUS Mode register, $C1 and the results of the conversion are presented in AuxADC Data C-BUS register $C9. This register also includes a bit field to indicate which input was selected when this conversion was executed. The AuxADC input can be routed to either of the RxIN signals, the SpareADC Input, the RSSI inputs or the AuxADC inputs under host control. In normal operation it is expected to be routed to one of the AuxADC Inputs.

### 7.4.12 Auxiliary DACs

The four Auxiliary DACs can be updated in any combination using the AuxDACWrite data task. In addition, DAC1 can be configured as a RAMDAC to output a series of values as part of the transmit timing sequence. The values and the rate at which they change are set up using config mode task.

### 7.4.13 Interrupt Operation

The CMX7032 will issue an interrupt on the IRQN line when the IRQ bit (bit 15) of the Status register and the IRQ Mask bit (bit 15) are both set to ‘1’. User Manual section 11.17 describes the situations which cause the IRQ bit to change from a ‘0’ to a ‘1’. The IRQN pin is an open collector output that requires an external pull-up resistor.

## 7.5 Operation of Tasks

This section describes modem and data tasks. Understanding their operation requires knowledge of the internal buffering of the CMX7032.

Tx and Rx data is double buffered. Each Tx or Rx channel has a Data Buffer. The host µC accesses the C-BUS registers and the modulator/demodulator directly accesses the Data Buffers. Tasks transfer data between the buffers and the C-BUS registers.

### 7.5.1 Tx Task Operation

Typical stages of Tx task operation are depicted in Figure 11 and occur as follows:

1. The host writes up to 4 words of data for transmission into the Write Data C-BUS registers.
2. The host writes the Command register, specifying a data task. This results in transfer of the data from the Write Data registers into the Tx Data Buffer.
3. Steps 1 and 2 can be repeated to load the Tx Data Buffer with a large block of data.
4. A Modem task can then be used to instruct the Tx Modulate pump to transmit the data in the Data Buffer. This causes the content of the Tx Data Buffer to be coded and CRC'd (if in burst mode) and transmitted to the MOD1 and MOD2 output pins.
5. Once the system is up and running any modem task may potentially take some time to execute as it may have to wait for the previous task to complete.

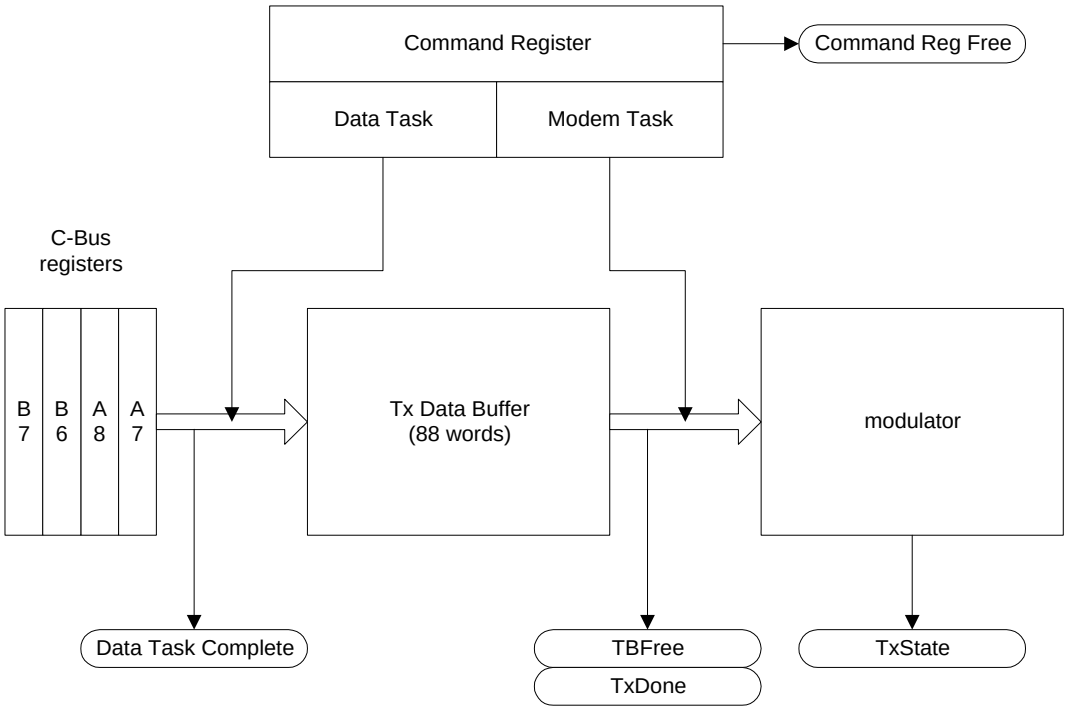



**Figure 11 Tx Task Operation**

### 7.5.2 Rx Task Operation

Typical stages of Rx task operation are depicted in Figure 12 and occur as follows:

1. A Modem task instructs the CMX7032 to transfer data from the Rx1/2 Modem to the Rx1/2 Data Buffer.
2. The host writes the Command register, specifying a Data task. This results in transfer of up to 4 words from the Rx1/2 Data Buffer into the Read Data C-BUS registers, from where it can be read by the host µC.
3. Steps 2 can be repeated (with host µC reads of the Read Data registers) to transfer a large block of data from the Rx1/2 Data Buffer to the host µC.
4. Once the system is up and running any modem task may take some time to execute, as it will need to wait for the data to be available from the modem.

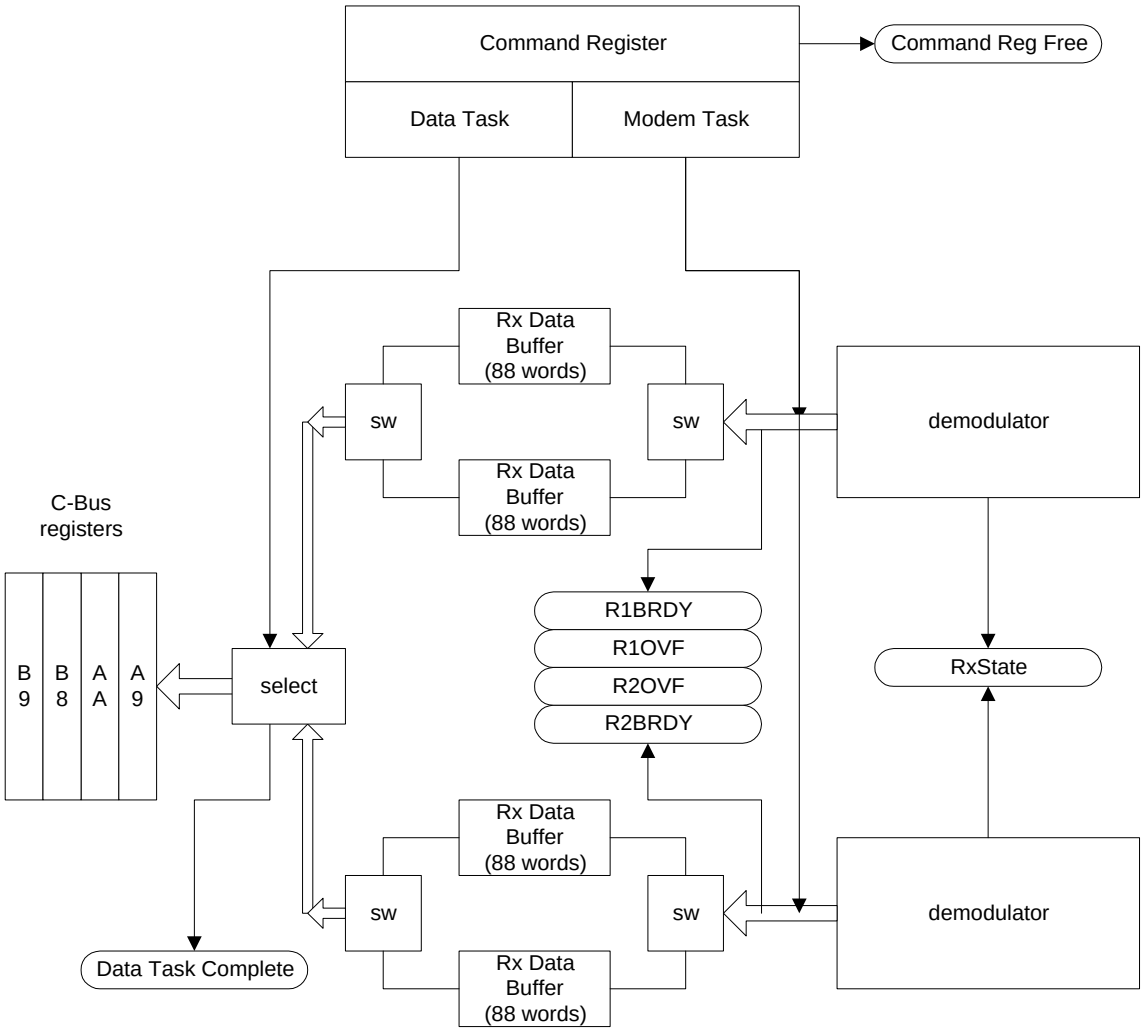



**Figure 12 Rx Task Operation**

### 7.5.3 Registers and Buffers for Tx/Rx Tasks

- **Command register**: contains Data and Modem task fields as described above.
- **Status register** – contains bits that indicate when tasks are complete, which can interrupt the host:
  - Command Reg Free
  - TBFREE
  - R1BRDY
  - R2BRDY
  - TxDONE
  - Rx1OVF
  - Rx2OVF
  - Config Task Complete
  - Data Task Complete
- **Interrupt Mask**: Host write register to specify which status bit can cause an interrupt.
- **Write Data registers** 0-3: Contain data written from host µC to transmit via the Tx Modulator.
- **Read Data registers** 0-3: Contain data received from Rx Demod for host µC to read.
- **Tx Data Buffer**: The Tx Data is double buffered, which allows the host µC to write to the Tx Data Buffer while the modulator is simultaneously transmitting data it reads from the Tx Modem Buffer. Each buffer is capable of holding one full (5-slot) AIS message.
- **Rx1/2 Data Buffer**: The demodulator writes data directly into these internal buffers. There are two buffers per channel which are used alternately every time a new burst is detected. This allows the host µC to read from one buffer while reception continues to fill the other. Each buffer is capable of holding a full (5-slot) AIS message.

### 7.5.4 Write Data Registers

An array of four, 16 bit, C-BUS write registers form the Write Data C-BUS registers.

The device reads and acts upon the content of these data write registers as instructed by the Data Task bits of the Command register while in transmit mode. Generally, they may be written at any time by the host µC with no effect on internal device operation. When a "Data task" is issued the Data registers will be read by the device and so should not be modified by the host µC until the Data Task complete bit is set in the Status register.

Data tasks access the registers as a number of words (1 to 4) or as a number of bits (1 to 16 in $A7), however if a bit-format Data Task is used it must be the final data task issued in a multi-data transfer from the host. The next data task issued should be a DataWordReset or DataBitReset to re-initialise the internal data buffer pointers (a bit-format task is usually used as the last transfer of a data block that is not a complete number of words in length).

Word-format:

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Register $A7 | Data write from host µC to device word 1(MSB sent first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $A8 | Data write from host µC to device word 2(MSB sent first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $B6 | Data write from host µC to device word 3(MSB sent first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $B7 | Data write from host µC to device word 4(MSB sent first) | | | | | | | | | | | | | | | |

Bit-format:

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Register $A7 | Data write from host µC to device bits 0-15, (bit 15 transmitted first) | | | | | | | | | | | | | | | |

### 7.5.5 Read Data Registers

An array of four, 16 bit, C-BUS read registers form the Read Data C-BUS registers.

The device writes into these registers as instructed by the Data Task bits of the Command register while in receive mode. The host µC can read these registers at any time except while a data task is in progress, in which case the host µC should wait until the Data task complete bit is set in the Status register.

Data tasks access the registers as a number of words (1 to 4) or as a number of bits (1 to 16 in $A7), however if a bit-format Data task is used it must be the final data task issued in a multi-data transfer to the host. The next data task issued should be a DataWordReset or DataBitReset to re-initialise the internal data buffer pointers (a bit-format task is usually used as the last transfer of a data block that is not a complete number of words in length).

Word-format:

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Register $A9 | Data read from device to host µC word 1 (MSB Rx first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $AA | Data read from device to host µC word 2 (MSB Rx first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $B8 | Data read from device to host µC word 3 (MSB Rx first) | | | | | | | | | | | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Register $B9 | Data read from device to host µC word 4 (MSB Rx first) | | | | | | | | | | | | | | | |

Bit-format:

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Register $A9 | Data read from device to host µC bits 0-15, (bit 15 received first) | | | | | | | | | | | | | | | |

### 7.5.6 Data Tasks

Data tasks are used to:

- Load data from the Write Data registers into Data Buffers while in normal or configuration modes.
- Load data from Data Buffers into the Read Data registers while in normal mode.
- Read, write or operate subsystems by passing data using the Write Data and Read Data registers.

**Table 3 Data Tasks**

| Name | Description |
|---|---|
| NULL | Null system task – takes no action. |
| DataWordReadNRx1 | • Copy N words (1 to 4) from Rx1 data buffer to C-BUS Read Data registers.<br>• Increment Rx1 data buffer pointer. |
| DataWordReadNRx2 | • Copy N words (1 to 4) from Rx2 data buffer to C-BUS Read Data registers.<br>• Increment Rx2 data buffer pointer. |
| DataWordWriteNTx | • Copy N words (1 to 4) from Write Data registers to Tx data buffer.<br>• Increment data buffer pointer. |
| DataWordResetNRx1 | • Reset Rx1 data buffer pointer to the top.<br>• Copy N words (1 to 4) from Rx1 data buffer to Read Data registers.<br>• Increment data buffer pointer. |
| DataWordResetNRx2 | • Reset Rx2 data buffer pointer.<br>• Copy N words (1 to 4) from Rx2 data buffer to Read Data registers.<br>• Increment data buffer pointer. |
| DataWordResetNTx | • Reset Tx data buffer pointer.<br>• Copy N words (1 to 4) from Write Data registers to Tx data buffer.<br>• Increment the data buffer pointer. |
| DataBitReadNRx1 | • Copy N bits (1 to 15) from Rx1 data buffer to Read Data register 0.<br>• Increment the data buffer pointer. |
| DataBitReadNRx2 | • Copy N bits (1 to 15) from Rx2 data buffer to Read Data register 0.<br>• Increment data buffer pointer. |
| DataBitWriteNTx | • Copy N bits (1 to 15) from Write Data register 0 to Tx data buffer.<br>• Increment data buffer pointer. |

| | |
|---|---|
| DataBitResetNRx1 | • Reset Rx1 data buffer pointer.<br>• Copy N bits (1 to 15) from Rx1 data buffer to Read Data register 0.<br>• Increment the data buffer pointer. |
| DataBitResetNRx2 | • Reset Rx2 data buffer pointer.<br>• Copy N bits (1 to 15) from Rx1 data buffer to Read Data register 0.<br>• Increment data buffer pointer. |
| DataBitResetNTx | • Reset Tx data buffer pointer.<br>• Copy N bits (1 to 15) from Write Data register 0 to Txdata buffer.<br>• Increment data buffer pointer. |
| AuxDACWrite | Interprets each of the first 1 to 4 words in the Write Data registers as a write command for the Auxiliary DACs |

### 7.5.7 Modem Tasks and Codes

Modem tasks transmit data on the MOD1 and MOD2 output pins or receive and recover data from the RxIN pins. Modem tasks also coordinate data transfer between the Data Buffers and their respective Modems.

Note that for receive tasks a 1 or 2 at the end of the task name refers to the Rx channel which is being addressed.

**Table 4 Modem Tasks**

| Name | Description | | |
|---|---|---|---|
| NULL | No command – takes no action. | | |
| Abort1<br>Abort2<br>AbortTx | Abort the ongoing modem task on Rx1, Rx2 or Tx. | | |
| ECM | Enter Configuration mode. | | |
| Rx Tasks | | | |
| Rx Raw bit = 0 | | Rx Raw bit = 1 or channel is set to DSC. | |
| RXB1<br>RXB2 | Rx Burst: Wait for a training sequence, then demodulate, decode and store the burst. | | |
| | | RXR1<br>RXR2 | Demodulate and store N words. N is defined by Rx data count register. |
| Tx Tasks | | | |
| Tx Raw bit = 0 | | Tx Raw bit = 1 | |
| TXB | Code and transmit AIS message using contents of data buffer. Start on next slot clock. | TDBS | Transmit contents of data buffer. Start on next slot clock. |
| | | TDB | Transmit N data bits from the Tx mod buffer. Start as soon as modulator is free. |
| | | PRBS | Transmit pseudorandom bit sequence. |
| | | TRW | Repeatedly transmit one word. |
| | | HCT | Hardware Control |

## 7.6 Transmission format

The CMX7032/CMX7042 is capable of transmitting AIS data in either raw mode or burst mode. AIS Carrier Sensing (CSTDMA) for Class B systems is supported.

In AIS raw mode, data is passed directly from the Tx Data Buffer to the GMSK modulator, so the µC will be responsible for sending any necessary training sequences and performing HDLC processing and NRZI coding.

In AIS burst mode, the CMX7032/CMX7042 uses an internal message buffer to assemble an entire message (up to 5 slots) to which it automatically adds the training sequence, start/stop flags, CRC, bit stuffing and NRZI coding prior to transmission.

After setting up the appropriate registers, transmission is initiated by issuing a Tx Burst or Tx Raw task.

### 7.6.1 Transmit Tasks

- **AbortTx:**
  This causes the current task on the Tx channel to abort. It also clears the Tx modem buffer.
- **TXB – Transmit AIS Burst**
  This task can only be executed if the Tx Raw bit (bit 5 in the command register) is cleared to '0'. This causes the CMX7032 to take the contents of the Tx Data buffer, apply AIS data coding and transmit the resulting AIS message. The transmit sequence will start on the next SlotCLK edge.

The following four transmit tasks can only be executed if the Tx Raw bit is set to '1'

- **TDBS – Transmit Data Buffer on SlotCLK**
  This causes the CMX7032 to transmit the data buffer contents using AIS modulation. No data coding is applied, the Transmit Sequence will start on the next SlotCLK edge.
- **TDB – Transmit Data Buffer**
  This causes the CMX7032 to transmit the data buffer contents using AIS modulation. No data coding is applied. The data will be transmitted as the modulator is available (Transmit Sequence is ignored).
- **PRBS – Transmit pseudorandom bit sequence**
  This task causes the CMX7032 to transmit an internally generated pseudorandom bit sequence. The sequence is 511 bits in length, but will repeat indefinitely until aborted using the Abort Tx task, (Transmit Sequence is ignored).
- **TRW – Transmit Repeated Word**
  This task causes the CMX7032 to repeatedly transmit the first word currently in the data buffer. Transmission will start immediately and will continue until an Abort Tx task is issued, (Transmit Sequence is ignored).
- **HCT – Hardware Control Task**
  Allows manual control of ancilliary hardware functions.

### 7.6.2 AIS Burst Mode Transmit

In AIS burst mode, the CMX7032/CMX7042 responds to a TXB task by performing bit stuffing, NRZI encoding and the addition of training sequence, start/stop flags and CRC checksum as required by AIS. Note: in AIS burst mode, the data words are automatically transmitted *least significant bit first* as required by the AIS specification.

A number of error conditions are checked for during AIS burst mode transmit, each of which causes transmission to be aborted and a Tx Done interrupt to be generated. The associated Tx States are:

- **Tx aborted, message too long:**
  This occurs if the internal message buffer is not big enough for the HDLC coded data (should not happen in normal operation, as the message buffer is big enough for a 5-slot message). This condition requires the µC to issue a AbortTx task.
- **Tx aborted, carrier sensed:**
  This occurs if the CSTDMA mechanism is enabled and the CS measurement has exceeded the CS Threshold. The data is retained so that it can be re-transmitted in a subsequent slot (by issuing another TXB task) should the host request it. The slot selection should follow the rules given in the relevant international standard.
- **Tx aborted, buffer not ready:**
  This occurs in burst mode if the internal data coding has not completed before the timing_start value expires.

### 7.6.3 Transmit Example

The following detailed example describes the process of loading and transmitting an AIS message in Burst mode.

**Table 5 AIS Burst Transmit Example**

| | Description | Cmd Reg Free | Data Task | TBFREE | TxDONE |
|---|---|---|---|---|---|
| 1. | The host should ensure that the TBFREE, Data Task and CmdReg Free bits are set. | 1 | 1 | 1 | 1 |
| 2. | The host loads the first N(typically 4) data words into the write data registers. | 1 | 1 | 1 | 1 |
| 3. | The host issues a DataWordResetNTx Data Task. | 0 | 1 | 1 | 1 |
| 4. | Device reads the command register & notes task types | 1 | 0 | 1 | 1 |
| 5. | Device carries out the data task by copying the N data words as the first N data words of the data buffer | 1 | 1 | 1 | 1 |
| 6. | The steps above may be repeated (Using DataWordWriteN tasks) to load many words until the data buffer contains enough data to carry out the desired modem task. | | | | |
| 7. | The host writes a TXB task to the command register to start the Tx process. | 0 | 1 | 1 | 1 |
| 8. | Device reads the Command register | 1 | 1 | 0 | 1 |
| 9. | Device codes the data. Tx state changes from Idle to Tx Pending | 1 | 1 | 1 | 0 |
| 10. | When the transmit point arrives (SlotCLK), the Tx State changes to *Tx in progress* and the TxSequence is activated. | | | | |
| 11. | The Tx Modem Buffer will gradually empty as the Tx Modulator continues transmitting. | 1 | 1 | 1 | 1 |
| 12. | When the transmission ends the TxDone bit in the Status register will be set, generating an interrupt if enabled. The host should then check the Tx state bits in the Status2 register to see if transmission was successful. | 1 | 1 | 1 | 1 |

Note that if CSTDMA mode is active and a carrier is sensed in the selected channel at the beginning of the requested transmit slot, the transmission is aborted (Tx State changes to *Tx aborted, carrier sensed*) – this causes a TxDone interrupt to be generated, however the data in the Tx Data Buffer is retained, so the µC can choose to issue an AbortTx task and clear the Tx Data Buffer, or reschedule the transmission in another slot.

### 7.6.4 AIS Raw Mode Transmit

In AIS raw mode, transmit data is passed directly from the Tx Data Buffer to the GMSK modulator. The µC must calculate the entire transmitted message including the training sequence, HDLC processing (start/stop flags, bit stuffing, and CRC insertion) and NRZI coding. Note: In AIS raw mode, data words written to the CMX7032 are transmitted *most significant bit first*. The AIS message structure, however, requires each message byte to be output *least significant bit first*. The µC must therefore ensure that during the process of HDLC processing and NRZI coding that the resulting data bytes are correctly reversed.

### 7.6.5 Transmitter Timing Control

The CMX7032/CMX7042 can be configured to control the timing of transmission events whenever a Tx Burst Modem task is executed. This includes the enabling of external RF circuits (e.g. synthesisers and power amplifier), as well as the time at which internal data modulation begins. The flexibility of this timing control allows the CMX7032/CMX7042 to be simply adapted to the characteristics of the RF transmit circuits. The control of the external RF transmit circuits is performed using the TxEnable pin and the AuxDAC1 ramping function.

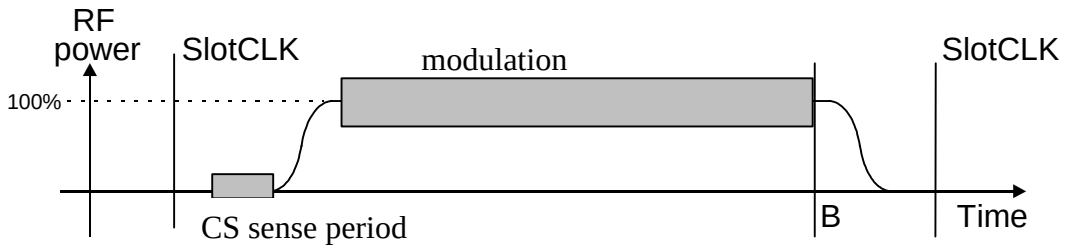



**Figure 13 Typical AIS Transmission**

A typical AIS transmission is shown in Figure 13. The CMX7032/CMX7042 starts timing relative to the rising edge of SlotCLK. At the end of a transmission, a sequence of "power-down" actions is performed which are timed relative to the last message bit having been modulated, shown as point B in Figure 13. In this way differences in message length due to bit stuffing are automatically accommodated.

The relative timings of the transmit sequence events are configured as a table of values that are loaded into the CMX7032/CMX7042 using a Config Task operation (User Manual section 11.15.7) – this operation **must** be performed before any transmissions are attempted. Typically, this will only need to be done once as part of an initialisation routine. All timings are measured in units of "ticks", each of which lasts for 1/24000Hz ( ≅ 41.666µs).

The transmit sequence consists of two initial setting values followed by a number of different event types. These are:

- Initial delay from the SlotCLK edge
- Active state of the TxEnable pin
- Changes to the external hardware, via the TxEnable pin (typically used to turn the Tx on/off) and the AuxDAC1 ramp up/down.
- Trigger for the start/end of the CSTDMA sensing period (if CSTDMA is enabled).
- Timing triggers for the start and end of the data modulation.
- A dummy event in case any of the above are not required in the application.

The transmit event sequence is programmed using a Config task, see User Manual section 11.15.7.

| b3 | b2 | b1 | b0 | Event id | description |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | dummy | Do nothing |
| 0 | 0 | 0 | 1 | CSTDMA_START | Defines the start of the CSTDMA sensing window |
| 0 | 0 | 1 | 0 | CSTDMA_END | Defines the end of the CSTDMA sensing window |
| 0 | 0 | 1 | 1 | Tx_ON | Pin TxEnable is asserted |
| 0 | 1 | 0 | 0 | RAMDAC_UP | AuxDAC1 will start executing a Ramp up |
| 0 | 1 | 0 | 1 | MODULATE_START | Defines the start of data modulation |
| 0 | 1 | 1 | 0 | MODULATE_END | Delay from the end of modulation (based on the last data bit loaded into modem - includes a 20 tick delay for the internal filters) |
| 0 | 1 | 1 | 1 | RAMDAC_DOWN | AuxDAC1 will start executing a Ramp down |
| 0 | 1 | 0 | 0 | Tx_OFF | Pin TxEnable is de-asserted |

**Table 6 Tx Sequence events**

When calculating the MODULATE_START timing value, the delay through the CMX7032/CMX7042's internal transmit filters and any external components must be taken into account to ensure that data bits appear on-air at the correct time (the filter delays are specified in section 8.1.4.The MODULATE_END event has an in-built delay of 20 ticks to allow the last bit to make its way out of the transmit filter. Allowance must be made for this built-in delay, as well as for the delay through any external components, when calculating the timing of the transmit power down events.

A working example of how to set up a transmit event sequence is shown in Table 7 (the order of events and delay timings shown are for illustrative purposes only):

| Parameter | event | delay | total | Explanation |
|---|---|---|---|---|
| CSTDMA_START | 1 | 10 | 20 | Insert 10 tick delay then start monitoring the chosen Rx input for a signal which may cause an abort (if CSTDMA enabled). |
| MODULATE_START | 5 | 17 | 37 | Insert 17 tick delay then start feeding data to the transmit modulator and filters (this allows for the 20 tick storage delay in the Tx filters so that modulated data appears at the end of the RAMDAC ramp_up period – tick 57). |
| CSTDMA_END | 2 | 10 | 47 | Insert 10 tick delay then stop CSTDMA monitoring. |
| TX_ON | 3 | 0 | 47 | Assert the TxEnable line (assuming not aborted) |
| RAMDAC_UP | 4 | 3 | 50 | Insert 3 tick delay then initiate the RAMDAC ramp-up (for AIS, the transmitted signal will be carrier only at this point) |
| *At this point during a transmission the CMX7032/CMX7042 feeds the entire message to the transmit modulator bit-by-bit. All subsequent transmit events are timed relative to the end of the last message bit, indicated by the MODULATE_END event.* | | | | |
| MODULATE_END | 6 | 0 | 0 | The 0 delay defines this reference point as the same as the output of the final data bit to the RF circuits (“B” in Figure 13). |
| RAMDAC_DOWN | 7 | 0 | 0 | Initiate the RAMDAC ramp-down immediately |
| TX_OFF | 8 | 8 | 8 | Insert 7 tick delay (to allow RAMDAC to fully ramp down) then release the TxEnable line. |

Notes:
1. It is essential that the CSTDMA, and MODULATE *START* events precede their associated *END* events, otherwise undesirable results will be obtained.
2. MODULATE_START must appear in the first group of timed events (table entries 1–5), MODULATE_END must appear in the final group (table entries 6-8).
3. It is feasible to place the RAMDAC_DOWN task *before* the MODULATE_END task if it is desired to continue modulation during the Ramp down period.

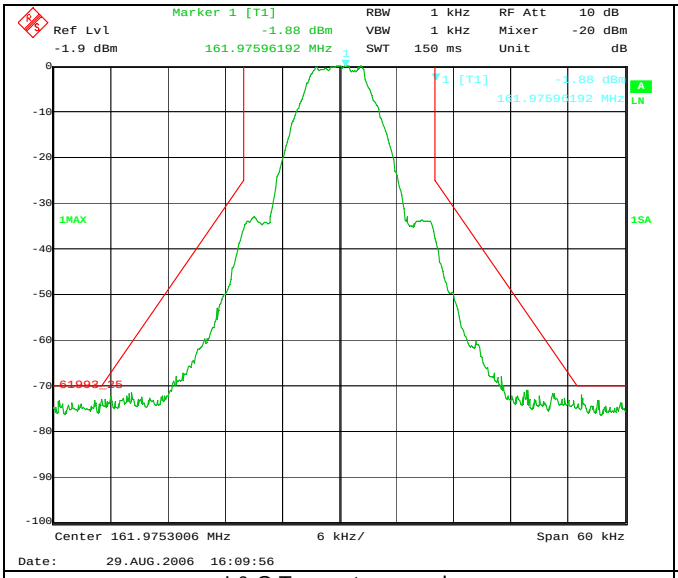
Assuming that the timing_start value has been set to 10 (see User Manual section 11.15.7) and the RAMDAC is set to its default values (312us), this sequence approximates to the Class B CSTDMA timing with ideal hardware (RAMDAC starts 20bits / 50 ticks after SlotCLK).

**Table 7 Example Tx Event Sequence Setup**

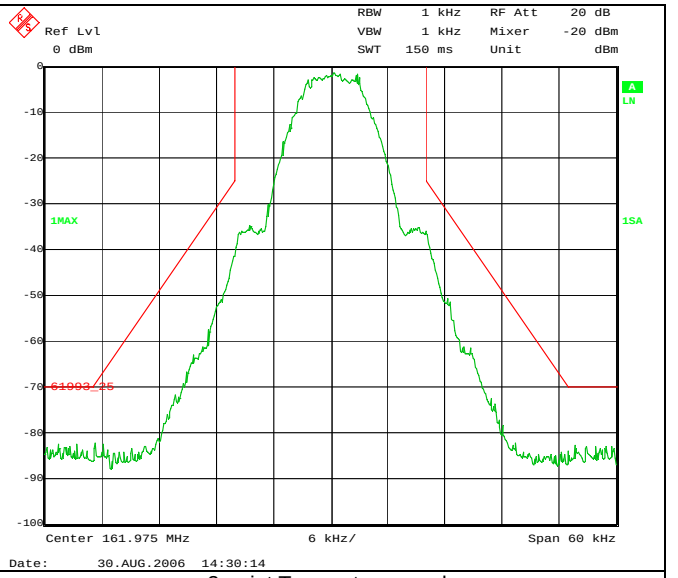
### 7.6.6 Modulation Formats

The CMX7032/CMX7042 can be configured to drive either a two-point VCO & Reference modulator or an I&Q modulator by selecting the appropriate Config task (see User Manual section 11.15.1).

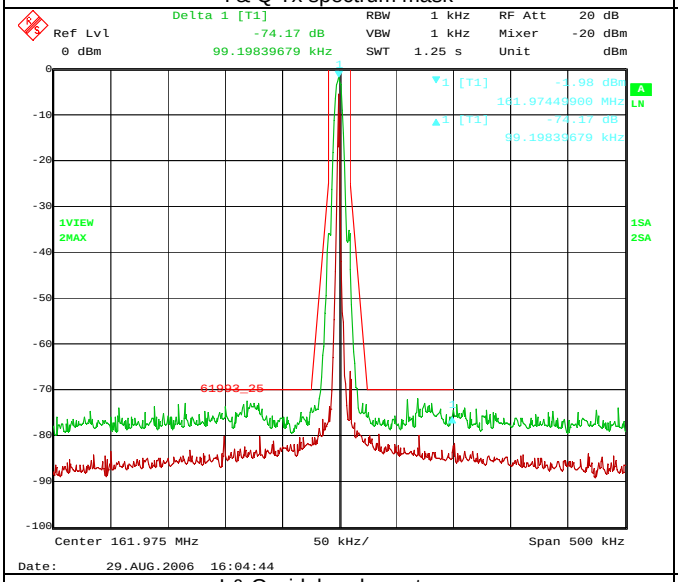
Typical Tx spectrum plots for both modes are shown below (generated by modulating a signal generator with the outputs of MOD1 and MOD2 and then analysing the signal on a spectrum analyser). Note that these plots represent the steady-state transmission and so are shown with the Class A and Class B-SOTDMA spectrum mask (-70dBc). The Class B–CSTDMA standard specifies a slotted transmission with a mask at –60dBc.



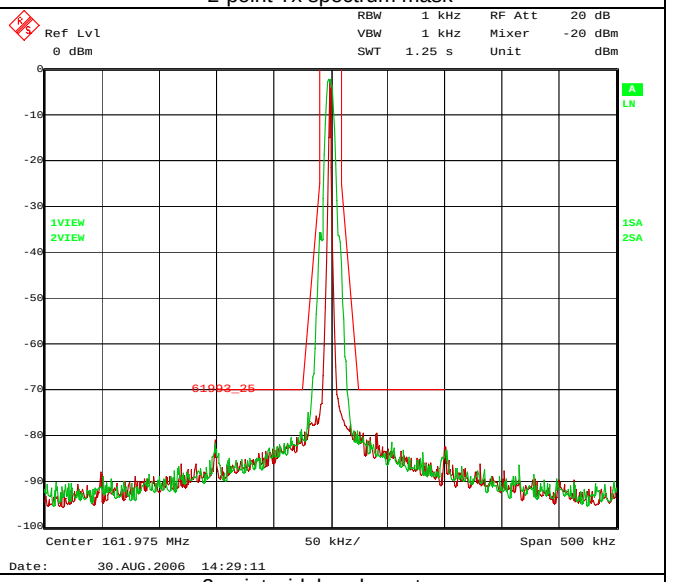

I & Q Tx spectrum mask




2-point Tx spectrum mask




I & Q wideband spectrum




2-point wideband spectrum

(lower trace shows un-modulated signal generator)

**Table 8 Tx spectrum masks**

## 7.7 Reception

The CMX7032/CMX7042 has two receive channels (Rx1 and Rx2) which are capable of receiving AIS data in either *raw* mode or *burst* mode, and either of which may be configured for DSC reception (FSK 1200 baud). The Rx1 and Rx2 channels can be configured and operated independently.

### 7.7.1 AIS Burst Mode Receive

The operation of receive channel Rx1 in AIS burst mode is described below (the operation of receive channel Rx2 in AIS burst mode is essentially identical to that of Rx1).

In AIS burst mode, once an RXB1 task has been issued, the Rx1 channel state changes to *Receiving* when a valid training sequence and start flag are detected. The CMX7032/CMX7042 then performs NRZI decoding and bit destuffing on the received data stream, and calculates the CRC checksum. Note: in AIS burst mode, the data words are automatically reversed so that they are presented to the host *most*

*significant bit first*. At the end of the message the receive channel state changes from *Receiving* to either *Idle* or one of four error states (below). At the same time, an “Rx State Alert” interrupt is flagged.

The four error conditions that the CMX7032/CMX7042 can detect in a received message (in burst mode) are:

- **Message too long or missing end flag.**
  This indicates that the received message, after bit de-stuffing, is too long to fit into the internal message buffer. This condition could be caused by a missing or corrupted end flag.
- **CRC mismatch.**
  This indicates that the received frame checksum does not match that calculated by the CMX7032/CMX7042, most probably as the result of one or more message bits being corrupted.
- **New frame header found when message buffer full.**
  This happens if the internal message buffers are still in use when another message arrives. This is caused by a failure of the host µC to read the received messages out quickly enough.
- **End flag not on byte boundary.**
  This indicates that the received message, after bit de-stuffing, is not a multiple of 8 bits. Assuming that the message was transmitted correctly, it is probably caused by an end flag being missed due to noise, and a subsequent message’s start flag being mis-identified as the expected end flag or a bit error causing the bit de-stuffer to fail.

If one of these four error conditions is detected in a received message the CMX7032/CMX7042 discards the message data.

If a message with no error is found the Rx1 channel state changes from *Receiving* to *Idle* (causing an “Rx State Alert” interrupt); the decoded message, comprising the burst information, three training sequence bytes, start flag, message payload, CRC bytes and end flag, is then copied to one of the CMX7032/CMX7042’s internal message buffers. When its turn comes around to be read out, it is copied to the Rx1 Data Buffer and an “R1BRDY” interrupt is generated. At this point the host can issue Read Data tasks to read back the burst and its associated parameters. Note: a new message will only generate an “R1BRDY” interrupt when any previous message has been read out from the Rx1 Data Buffer in its entirety.

The Rx1 channel state will stay in *Idle* until another RXB task is issued.

For any particular message, the three received (NRZI-decoded) training bytes in AIS burst mode will all be either $55 or $AA depending on the configuration of the remote transmitter, although the first few bits may be corrupted depending on the power-up characteristics of the remote transmitter and local receiver circuits.

The host must read the Rx data buffers sufficiently quickly to avoid an overflow condition occurring. This is only likely in a very heavily loaded AIS network. The worst case would involve the reception of a 5-slot burst followed by a single slot and then a third burst in contiguous slots. In this case the host would need to read the entire 5-slot burst out of the Rx Data buffer during reception of the single slot burst, such that the buffer is then available for the third burst in the sequence. This is further compounded by the need to monitor both Rx channels. Single slot AIS messages contain 168bits of data, which can be read by the host in 3 C-BUS RxData Read tasks. The maximal length 5-slot message contains upto 840 bits which can be read by the host in 15 C-BUS RxData Read tasks during the 26ms of a single slot. This implies that the C-BUS must be running at a speed greater than 128kHz.

### 7.7.2 AIS Raw Mode Receive

The operation of receive channel Rx1 in AIS raw mode is described below (the operation of receive channel Rx2 in AIS raw mode is essentially identical to that of Rx1, but is controlled through its own set of tasks). Note that both channels operate in either Raw or Burst mode, it is not possible to select AIS Raw mode on one channel and AIS Burst mode on the other.

In AIS raw mode, the RxRAW bit in the Command register, $C8 must be set. The CMX7032/CMX7042 then searches the Rx1 channel for a header (training + start flag sequence) to detect the start of a message, then transfers the three training bytes and the start flag to the Rx1 Data buffer. The number of subsequent words transferred to the buffer is set by the value in the RxData Count register, which is read whenever the RXR task is executed. R1BRDY bit will be set when the programmed number of words have been transferred. The data can then be read back using Data Read tasks. A new RXR task should then be executed if it is required to recover further data words. It is the responsibility of the µC to perform all HDLC/NRZI decoding, CRC checking and end flag detection. The demodulated byte stream continues even after the end of a message and in the absence of a received signal (the data will then be indeterminate. Reception can be halted by issuing an Abort1 task.

Bit ordering of the received data in AIS raw mode is the same as in Tx AIS raw mode, i.e. the received bits are packed into words *most significant bit first*. As the AIS message structure requires message bytes to be transmitted least significant bit first, the µC must ensure that during the process of HDLC/NRZI decoding that the resulting data are correctly reversed. Depending on the configuration of the remote transmitter, one of four different types of NRZI encoded training bytes may be received – this situation arises because the AIS specification allows a transmitter's NRZI encoder to start in either of its two quiescent states, and the pre-NRZI encoded training bytes can also be one of two different types ($55 or $AA). Therefore, for any particular message, the three received training bytes in AIS raw mode will all be either $33, $66, $99 or $CC, although the first few bits may be corrupted depending on the power-up characteristics of the remote transmitter and local receiver circuits.

In AIS raw mode, whenever an Rx1 state reset is performed (by issuing an Abort1 task) the channel state becomes *Idle.* This changes to *Receiving* when the first valid training sequence and start flag have been detected, where it remains until another Rx1 state reset occurs.

In Raw mode a single Rx Data buffer is used, however it is 4 words longer than the equivalent Burst mode buffer as it does not contain any burst information.

### 7.7.3 DSC Receive

Either the Rx1 or Rx2 channel can be configured for DSC reception. The CMX7032/CMX7042 first applies 6dB/octave de-emphasis to the received signal, then demodulates the resulting 1200 baud NRZ FSK data. Only one of the channels at a time can be configured for DSC reception. The received data is packed into 16-bit words for onward transmission to the µC. The CMX7032/CMX7042 makes no attempt to perform dot pattern or data phasing detection, those functions must be performed by the host µC. No attempt is made to correctly align data, it is simply packed into words (most significant bit first) as it arrives.

### 7.7.4 Receive Example

The following detailed example describes the process of receiving an AIS message in Burst mode, on Rx channel 1.

**Table 9 AIS Burst Receive Example**

| | Description | Cmd Reg Free | DataTask | R1BRDY | Rx1B OVF |
|---|---|---|---|---|---|
| 1 | The host should ensure that the R1BRDY, Data Task and CmdReg Free bits are set. | 1 | 1 | 1 | 0 |
| 2 | The host issues an RXB modem task to receive an AIS burst. Note that the Rx Burst mode bit must be set. | 0 | 1 | 1 | 0 |
| 3 | Device executes the modem task | 1 | 1 | 0 | 0 |
| 4 | Device waits for a training sequence and start flag, then begins to demodulate and decode data. The Rx state changes from Idle to receiving. If the burst has a correct CRC, the result is placed in the data buffer, if not, it is discarded. An Rx state alert is issued and the Rx state changes from receiving to idle, or an error condition. | 1 | 1 | 1 | 0 |
| 5 | The host issues a DataWordResetNRx1 Data Task . | 0 | 1 | 1 | 0 |
| 6 | Device reads the command register & notes task types | 1 | 0 | 1 | 0 |
| 7 | Device carries out the data task by copying the first N data words of the data buffer into the Read Data registers. | 1 | 1 | 1 | 0 |
| 8 | The above 3 steps can be repeated (Using DataWordN tasks) to read many words out of the device. | | | | |

### 7.7.5 Rx Tasks

- **Abort Rx1/Rx 2**
  This causes the current task on the specified Rx channel to abort. It also clears the modem buffer of the specified Rx channel.

- **RxB1/2 – Receive Burst on Rx1/Rx2**
  This causes the specified Rx channel to wait for a good training sequence and start flag, then demodulate, decode and store the burst in its data buffer. See section 7.7.1 for a detailed discussion of AIS burst reception. The recovered data can be read by the host issuing an appropriate Data Task. The first four words of the buffer contain the burst information:
  Word1 ($A9) – Burst length in bytes.
  Word2 ($AA) – Time of arrival, measured in 24kHz ticks from the SlotCLK.
  Word3 ($B8) – Value proportional to DC offset of Rx Input.
  Word4 ($B9) – reserved
  Note: RXB tasks can only be issued when the Rx Raw bit is cleared to ‘0’.

- **RxR1/2 – Receive N Raw words on Rx1/Rx2**
  This causes the specified Rx channel to wait for a good training sequence and start flag, then demodulate and store N words in its Data buffer. N is specified in the Rx data count register. Burst information is NOT recorded in Raw mode, so there are an additional 4 words available in the Data Buffer compared with Burst mode operation.
  Note: RxR tasks can only be issued when the Rx Raw mode bit is set to ‘1’.

### 7.8 Configuration Tasks and Codes

The device executes Configuration Tasks while in configuration mode. (See section 7.4.2 for a description of device operating modes and how to change between them, and User Manual section 11.15 for more details on a particular task). These tasks and their data are used to configure device subsystems.

Data required for the Config task is loaded into the device using a Data Task, which can be executed at the same time as the Config task if it requires less than four words.

**Table 10 Configuration Tasks**

| Config Task | words | description | User Manual section |
|---|---|---|---|
| NULL | 0 | Do nothing | |
| EXIT_CONFIG | 0 | Return to Normal Mode | |
| Tx I/Q or 2-point | 1 | Sets MOD1 and MOD2 output format (2-point or I and Q) | 11.15.1 |
| Tx MOD levels | 1 | Sets output levels on MOD1 and MOD2 signal pins | 11.15.2 |
| RSSI_lookup | 128 | Loads RSSI calibration data | 11.15.3 |
| RSSI_window | 2 | Sets RSSI window timing and length | 11.15.4 |
| RSSI_gain | 1 | Sets RSSI gain factor | 11.15.5 |
| CSTDMA_gain | 1 | Sets CSTDMA gain factor | 11.15.6 |
| Tx_sequence | 18 | Loads Tx sequence commands | 11.15.7 |
| Sleep Timing | 3 | Sets Rx Sleep timing | 11.15.7 |
| RAMDAC load | 1 - 68 | Configures RAMDAC and loads data table | 11.15.9 |
| Device Ident | 2 | Reads back the Device Ident and Version number | 11.15.10 |
| Reserved | | | |
| Reserved | | | |
| Reserved | | | |
| Reference clock | 1 | 0 = 19.2MHz, 1 = 9.6MHz | 11.15.10 |
| Reserved | | | |

### 7.9 RF Synthesizer (CMX7032 only)

The CMX7032 includes two Integer-N RF synthesizers, each comprising a divider, phase comparator and charge pump. The divider has two sets of N and R registers: one set can be used for transmit and the other for receive. The division ratios can be set up in advance by means of C-BUS registers. Then, a single C-BUS command will change over from the transmit to the receive division ratios, or vice versa, enabling a fast turnaround.

External RF components are needed to complete the synthesizer circuit. A typical schematic for one synthesizer, with external components, is shown in Figure 14.

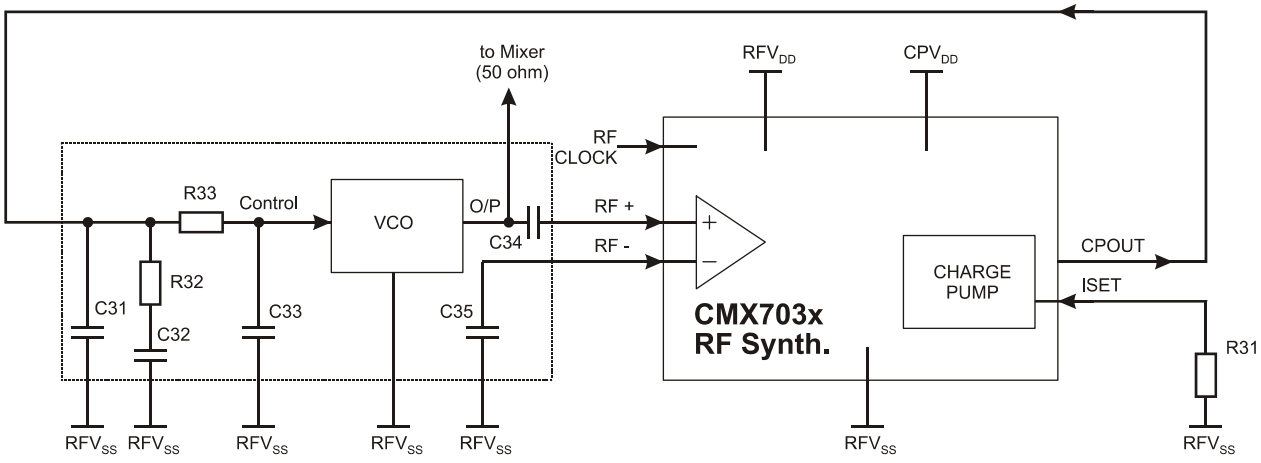



**Figure 14 Example RF Synthesizer Components**

| | | | |
|---|---|---|---|
| R31 | 0Ω | C31 | 820pF |
| R32 | 18kΩ | C32 | 8.2nF |
| R33 | 18kΩ | C33 | 680pF |
| | | C34 | 1nF |
| | | C35 | 1nF |

Resistors ±5%, capacitors and inductors ±20% unless otherwise stated.

Note: R31 is chosen within the range 0Ω to 30kΩ and selects the nominal charge pump current.

It is recommended that C34 and C35 are kept close to the VCO and that the stub from the VCO to the CMX7032 is kept as short as possible. The loop filter components should be placed close to the VCO.

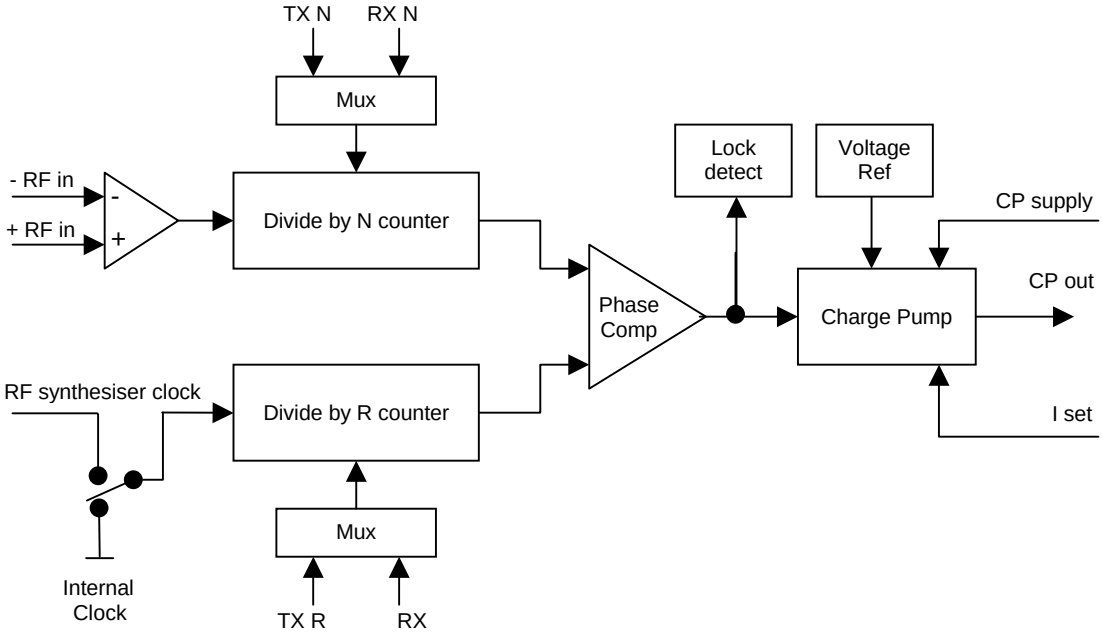



**Figure 15 Single RF Channel Block Diagram**

The two RF synthesizers are programmable to any frequency in the range 100MHz to 600MHz. Figure 15 is a block diagram of one synthesizer channel. The RF synthesizer clock is the same 9.6MHz or 19.2MHz clock as is used by the baseband circuitry. The RF synthesizer clock is common to both channels. The charge pump supply (CP supply, $CPV_{DD}$) is also common to both channels. The +/-RF in pins, CPout, Iset and $RFV_{SS}$ pins are channel specific and designated as either RF1+, RF1-, CP1OUT, ISET1, RFVSS or RF2+, RF2-, CP2OUT, ISET2, RFVSS on the Signal List in section 3. The N and R values for Tx and Rx modes are channel specific and can be set from the host µC via the C-BUS. Various channel specific status signals are also accessible via C-BUS. The divide by N counter is 20 bits; the R counter is 13 bits. Typical external components are shown in Figure 14.

Both synthesizers are phase locked loops (PLLs) of the same design, utilising external VCOs and loop filters. The VCOs need to have good phase noise performance although it is likely that the high division ratios used will result in the dominant noise source being the reference oscillator. The phase detectors are of the phase-frequency type with a high impedance charge pump output requiring just passive components in the loop filter. Lock detect functions are built in to each synthesizer and the status reported via C-BUS. A transition to out-of-lock can be detected and communicated via a C-BUS interrupt to the

host µC. This can be important in ensuring that the transmitter cannot transmit in the event of a fault condition arising.

Two levels of charge pump gain are available to the user, to facilitate the possibility of locking at different rates under program control. A current setting resistor (R31) is connected between the ISET pin (one for each PLL system) and the respective $RFV_{SS}$. This resistor will have an internally generated band gap voltage expressed across it and may have a value of 0Ω to 30kΩ, which (in conjunction with the on-chip series resistor of 9.6kΩ) will give charge pump current settings over a range of 2.5mA down to 230µA (including the control bit variation of 4 to 1). The value of the current setting resistor (R31) is determined in accordance with the following formulae:

gain bit set to 1: R31 (in Ω) = (24/Icp) – 9600
gain bit cleared to 0: R31 (in Ω) = (6/Icp) – 9600
where Icp is the charge pump current (in mA).

Note that the charge pump current should always be set to at least 230µA. The 'gain bit' refers to either bit 3 or bit 11 in the RF Channel Control register, $B3.
The step size (comparison frequency) is programmable; to minimise the effects of phase noise this should be kept as high as possible. This can be set as low as 2.5kHz (for a reference input of 20MHz or less), or up to 200kHz – limited only by the performance of the phase comparator.

The frequency for each synthesizer is set by using two registers: an 'R' register that sets the division value of the input reference frequency to the comparison frequency (step size), and an 'N' register that sets the division of the required synthesized frequency from the external VCO to the comparison frequency. This yields the required synthesized frequency (Fs), such that:

$Fs = (N / R) \times F_{REF}$ where $F_{REF}$ is the selected reference frequency

Other parameters for the synthesizers are the charge pump setting (high or low)

- Since the set-up for the PLLs takes 4 x "RF Channel Data register" writes it follows that, while updating the PLL settings, the registers may contain unwanted or intermediate values of bits. These will persist until the last register is written. It is intended that users should change the content of the "RF Channel Data register" on a PLL that is disabled, powersaved or selected to work from the alternate register set ("Tx" and "Rx" are alternate register sets). There are no interlocks to enforce this intention. The names "Tx" and "Rx" are arbitrary and may be assigned to other functions as required. They are independent sets of registers, one of which is selected to command each PLL by changing the settings in the RF Channel Control register, $B3.

For optimum performance, a common master clock should be used for the RF synthesizers (RF Clock) and the baseband sections (Main and Auxiliary System Clocks). Using unsynchronised clocks can result in spurious products being generated in the synthesizer output and in some cases difficulty may be experienced with obtaining lock in the RF synthesizers.

**Lock Status**
The lock status can be observed by reading the RF Channel Status register, $B4, and the individual lock status bits can (subject to masking) provide a C-BUS interrupt.

The lock detector can use a tolerance of one cycle or four cycles of the reference clock (not the divided version that is used as a comparison frequency) in order to judge phase lock. An internal shift register holds the last three lock status measurements and the lock status bits are flagged according to a majority vote of these previous three states. Hence, one occasional lock error will not flag a lock fail. At least two successive phase lock events are required for the lock status to be true. Note that the lock status bits confirm phase lock to the measured tolerance and not frequency lock. The synthesizer may take more time to confirm phase lock with the lock status bits than the time to switch from channel to channel. The purpose of a 4-cycle tolerance is for the case where a high frequency reference oscillator would not forgive a small phase error.

**RF Inputs**
The RF inputs are differential and self biased (when not powersaved). They are intended to be capacitatively coupled to the RF signal. The signal should be in the range 0dBm to –20dBm (not necessarily balanced). To ensure an accurate input signal the RF should be terminated with 50Ω as close to the chip as possible and with the "+" and "–" inputs capacitatively coupled to the input and ground, keeping these connections as short as possible. The RF input impedance is almost purely capacitative and is dominated by package and printed circuit board parasitics.

**Guidelines for using the RF Synthesizers**
- RF input slew rate (dv/dt) should be 14 V/µs minimum.
- The RF Synthesizer 2.5V digital supply can be powered from the VDEC output pin.
- RF clock sources and other, different clock sources must not share common IC components, as this may introduce coupling into the RF. Unused ac-coupled clock buffer circuits should be tied to a dc supply, to prevent them oscillating. By default the RF clock source is routed to the XTAL / Oscillator input internally.
- It is recommended that the RF Synthesizers are operated with maximum gain Iset (ie. Iset tied to RFVss).
- The loop components should be optimised for each VCO.

### 7.10 System Clock Synthesizers
Two System Clock outputs, SysCLK1and SysCLK2, are available to drive additional circuits, as required. These are digital phase locked loop (PLL) clocks that can be programmed via the System Clock registers with suitable values chosen by the user. The System Clock PLL Configure registers ($AB and $AD) control the values of the VCO Output divider and Main Divide registers, while the System Clock Ref. Configuration registers ($AC and $AE) control the values of the Reference Divider and signal routing configurations. The PLLs are designed for a reference frequency of 96kHz.

The System Clock output divider stages are designed so that it has an 1:1 Mark-to-Space ratio when an even divide number is selected.

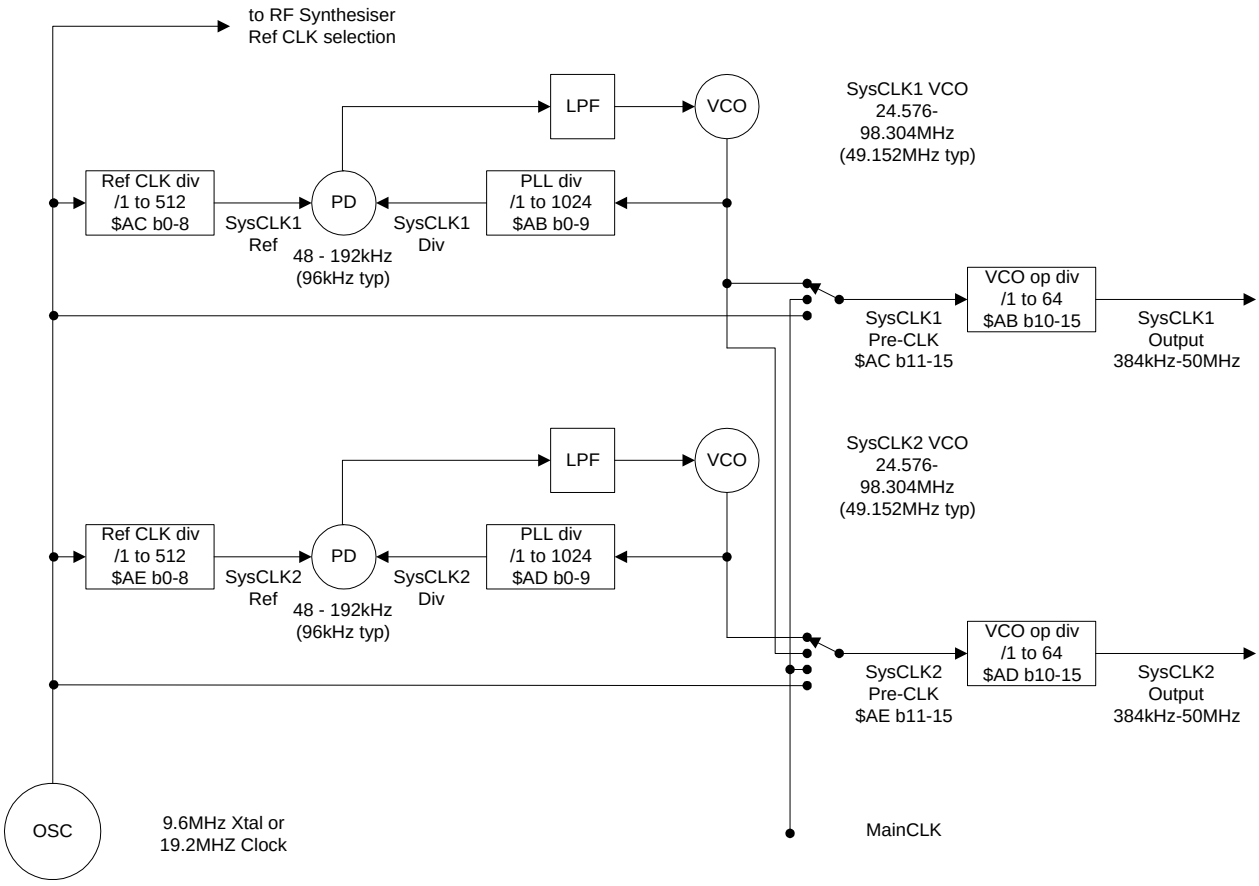



**Figure 16 System Clock Generation**

The CMX7032 includes a 2-pin crystal oscillator circuit. This can either be configured as a 9.6MHz xtal oscillator, or the XTAL input can be driven by an externally generated 19.2MHz clock.

### 7.11 Powersave

The CMX7032 implements a comprehensive power-saving scheme which will automatically enable the sections of the device that are required and return them to their power-saved state when no longer needed. In addition, a user defined Sleep mode maybe enabled which allows the device to drop into power-saving mode should a Receiver channel not detect any activity at the start of the AIS burst. This feature should be used with care to ensure that any peripheral circuits are powered-up again in time to receive the following burst correctly.

When an Rx channel enters or leaves Sleep mode, an Rx State IRQ may be asserted, which allows the host to powersave other external circuits under its control.

## 7.12 C-BUS Register Summary

**Table 11 C-BUS registers**

| ADDR. (hex) | | REGISTER | Word Size (bits) |
|---|---|---|---|
| $01 | W | C-BUS RESET | 0 |
| | | | |
| $A7 | W | TxData Write1 | 16 |
| $A8 | W | TxData Write 2 | 16 |
| $A9 | R | RxData Read 1 | 16 |
| $AA | R | RxData Read 2 | 16 |
| $AB | W | System Clk 1 PLL configure | 16 |
| $AC | W | System Clk 1 Ref configure | 16 |
| $AD | W | System Clk 2 PLL configure | 16 |
| $AE | W | System Clk 2 Ref configure | 16 |
| $AF | | Reserved | |
| | | | |
| $B0 | | Reserved | 16 |
| $B1 | W | Input Gain and Signal Routing | 16 |
| $B2 | W | RF Channel Data | 16 |
| $B3 | W | RF Channel Control | 16 |
| $B4 | R | RF Channel Status | 8 |
| $B5 | W | Rx Raw Data Counter | 16 |
| $B6 | W | TxData Write 3 | 16 |
| $B7 | W | TxData Write 4 | 16 |
| $B8 | R | RxData 3 / Checksum 1 high | 16 |
| $B9 | R | RxData 4 / Checksum 1 low | 16 |
| $BA | R | RSSI1 | 16 |
| $BB | R | RSSI2 | 16 |
| $BC | | Reserved | |
| $BD | | Reserved | |
| $BE | | Reserved | |
| $BF | | Reserved | |
| | | | |
| $C0 | | Reserved | 16 |
| $C1 | W | Mode | 16 |
| $C2 | W | CS Threshold | 16 |
| $C3 | | Reserved | |
| $C4 | | Reserved | |
| $C5 | R | Status 2 | 16 |
| $C6 | R | IRQ Status | 16 |
| $C7 | | Reserved | |
| $C8 | W | Command | 16 |
| $C9 | R | AuxADC Data | 16 |
| $CA | | Reserved | 16 |
| $CB | | Reserved | 16 |
| $CC | | Reserved | 16 |
| $CD | | Reserved | 16 |
| $CE | W | Interrupt Mask | 16 |
| $CF | | Reserved | |

All other C-BUS addresses (including those not listed above) are either reserved for future use or allocated for production testing and must not be accessed in normal operation.

# 8 Performance Specification

## 8.1 Electrical Performance

### 8.1.1 Absolute Maximum Ratings

Exceeding these maximum ratings can result in damage to the device.

| | Min. | Max. | Unit |
|---|---|---|---|
| Supply: $DV_{DD}$- $DV_{SS}$ | –0.3 | 4.5 | V |
| $AV_{DD}$- $AV_{SS}$ | –0.3 | 4.5 | V |
| $RFV_{DD}$- $RFV_{SS}$ | –0.3 | 4.5 | V |
| $CPV_{DD}$- $RFV_{SS}$ | –0.3 | 4.5 | V |
| Voltage on any pin to $DV_{SS}$ | –0.3 | $DV_{DD}$ + 0.3 | V |
| Voltage on any pin to $AV_{SS}$ | –0.3 | $AV_{DD}$ + 0.3 | V |
| Voltage on any pin to $RFV_{SS}$ (excluding $CPV_{DD}$) | –0.3 | $RFV_{DD}$ + 0.3 | V |
| Current into or out of any power supply pin (excluding $V_{BIAS}$) (i.e. $V_{DEC}$, $AV_{DD}$, $AV_{SS}$, $DV_{DD}$, $DV_{SS}$, $CPV_{DD}$, $RFV_{DD}$ or $RFV_{SS}$) | –30 | +30 | mA |
| Current into or out of any other pin | –20 | +20 | mA |
| Voltage differential between power supplies: | | | |
| $DV_{DD}$ and $AV_{DD}$ or $CPV_{DD}$ | 0 | 0.3 | V |
| $AV_{DD}$ and $CPV_{DD}$ | 0 | 0.3 | V |
| $DV_{SS}$ and $AV_{SS}$ or $RFV_{SS}$ | 0 | 50 | mV |
| $AV_{SS}$ and $RFV_{SS}$ | 0 | 50 | mV |

| **All Packages** | **Min.** | **Max.** | **Unit** |
|---|---|---|---|
| Storage Temperature | –55 | +125 | °C |
| Operating Temperature | –40 | +85 | °C |

| **Q1 Package (64-pad VQFN) or Q3 Package (48-pad VQFN)** | **Min.** | **Max.** | **Unit** |
|---|---|---|---|
| Total Allowable Power Dissipation at Tamb = 25°C | – | 1410 | mW |
| ... Derating | – | 14 | mW/°C |
| **L9 Package (64-pin LQFP) or L4 Package (48-pin LQFP)** | **Min.** | **Max.** | **Unit** |
| Total Allowable Power Dissipation at Tamb = 25°C | – | 668 | mW |
| ... Derating | – | 6 | mW/°C |

### 8.1.2 Operating Limits

Correct operation of the device outside these limits is not implied.

| | Notes | Min. | Max. | Unit |
|---|---|---|---|---|
| Supply Voltage: | | | | |
| $DV_{DD}$ – $DV_{SS}$ | | 3.0 | 3.6 | V |
| $AV_{DD}$ – $AV_{SS}$ | | 3.0 | 3.6 | V |
| $CPV_{DD}$ – $RFV_{SS}$ | | 3.0 | 3.6 | V |
| $RFV_{DD}$ – $DV_{SS}$ | 1 | 2.25 | 2.75 | V |
| $V_{DEC}$ – $DV_{SS}$ | 2 | 2.25 | 2.75 | V |
| Operating Temperature | | –40 | +85 | °C |
| Clock Frequency | | 9.6 | 19.2 | MHz |
| Function Image size | | 24 | 46 | kBytes |

**Notes:** 1 The $V_{DEC}$ supply is automatically created from $DV_{DD}$ by the on-chip voltage regulator.

2 The $RFV_{DD}$ supply can be supplied from the $V_{DEC}$ supply, if preferred.

### 8.1.3 Operating Characteristics

For the following conditions unless otherwise specified:

External components as recommended in **Figure 2**.
Maximum load on digital outputs = 30pF.
Clock Frequency =19.2MHz (±20ppm); Tamb = –40°C to +85°C.
$AV_{DD}$ = $DV_{DD}$ = $CPV_{DD}$ = 3.0V to 3.6V; $RFV_{DD}$ = 2.25V to 2.75V.
Reference Signal Level = 300mV pk-pk with $AV_{DD}$ = 3.3V.
Signal levels track with supply voltage, so scale accordingly.
Signal to Noise Ratio (SNR) in bit rate bandwidth.
Input stage gain =0dB. Output stage attenuation = 0dB.

| DC Parameters | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **Supply Current** | 21 | | | | |
| **All Powersaved (Config mode)** | | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 8 | 100 | µA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 4 | 20 | µA |
| $RFI_{DD}$ ($CPV_{DD}$ = 3.3V, $RFV_{DD}$ = 2.5V) | | – | 4 | 20 | µA |
| **Rx Mode** | 22 | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 14.4 | TBD | mA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 5.8 | TBD | mA |
| **Rx Mode (Sleep Enabled)** | 22 | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 10 | TBD | mA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 1.8 | TBD | mA |
| **Tx Mode** | 22 | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 20 | TBD | mA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 11 | TBD | mA |
| **Additional current for RF Synthesiser** | 23 | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 0 | – | mA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 0 | – | mA |
| $RFI_{DD}$ ($CPV_{DD}$ = 3.3V, $RFV_{DD}$ = 2.5V) | | – | 2.5 | 4.5 | mA |
| **Additional current for Auxiliary System Clock (output running at 4MHz)** | | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 250 | – | µA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 300 | – | µA |
| **Additional current for Auxiliary ADC** | | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 50 | – | µA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 1 | – | µA |
| **Additional current for each Auxiliary DAC** | | | | | |
| $DI_{DD}$ ($DV_{DD}$ = 3.3V, $V_{DEC}$ = 2.5V) | | – | 0 | – | mA |
| $AI_{DD}$ ($AV_{DD}$ = 3.3V) | | – | 200 | – | µA |
| **CLK** | 25 | | | | |
| Input Logic ‘1’ | | 70% | – | – | $DV_{DD}$ |
| Input Logic ‘0’ | | – | – | 30% | $DV_{DD}$ |
| Input current (Vin = $DV_{DD}$) | | – | – | 40 | µA |
| Input current (Vin = $DV_{SS}$) | | –40 | – | – | µA |
| **C-BUS Interface and Logic Inputs** | | | | | |
| Input Logic ‘1’ | | 70% | – | – | $DV_{DD}$ |
| Input Logic ‘0’ | | – | – | 30% | $DV_{DD}$ |
| Input Leakage Current (Logic ‘1’ or ‘0’) | | –1.0 | – | 1.0 | µA |
| Input Capacitance | | – | – | 7.5 | pF |

| DC Parameters | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **C-BUS Interface and Logic Outputs** | | | | | |
| Output Logic '1', ($I_{OH}$ = 120µA) | | 90% | – | – | $DV_{DD}$ |
| Output Logic '1', ($I_{OH}$ = 1mA) | | 80% | – | – | $DV_{DD}$ |
| Output Logic '0', ($I_{OL}$ = 360µA) | | – | – | 10% | $DV_{DD}$ |
| Output Logic '0', ($I_{OL}$ = -1.5mA) | | – | – | 15% | $DV_{DD}$ |
| "Off" State Leakage Current | | – | – | 10 | µA |
| IRQN (Vout = $DV_{DD}$) | | –1.0 | – | +1.0 | µA |
| REPLY_DATA (output HiZ) | | –1.0 | – | +1.0 | µA |
| **$V_{BIAS}$** | 26 | | | | |
| Output voltage offset wrt $AV_{DD}$/2 ($I_{OL}$ < 1µA) | | –2% | – | +2% | $AV_{DD}$ |
| Output impedance | | – | 22 | – | kΩ |

| AC Parameters | | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| **CLK Input** | | | | | | |
| 'High' pulse width | | 31 | 19 | – | – | ns |
| 'Low' pulse width | | 31 | 19 | – | – | ns |
| Input impedance (at 19.2MHz) | | | | | | |
| Powered-up | Resistance | | – | 150 | – | kΩ |
| | Capacitance | | – | 20 | – | pF |
| Powered-down | Resistance | | – | 300 | – | kΩ |
| | Capacitance | | – | 20 | – | pF |
| Clock frequency | | | - | 19.2 | – | MHz |
| Clock stability/accuracy | | | – | – | ±20 | ppm |
| Clock start up (from powersave) | | | – | 400 | – | ms |
| **$V_{BIAS}$** | | | | | | |
| Start up time (from powersave) | | | – | 30 | – | ms |
| **RxIN, SpareADC input** | | | | | | |
| Input impedance | | 34 | – | 1 | – | MΩ |
| Input signal range | | 35 | 10 | – | 90 | %$V_{DD}$ |
| Input signal envelope | | | 0.3 | | 2.2 | Vp-p |
| Load resistance (feedback pins) | | | 80 | – | – | kΩ |
| Amplifier open loop voltage gain (I/P = 1mV rms at 100Hz) | | | – | 60 | – | dB |
| Unity gain bandwidth | | | – | 1.0 | – | MHz |
| **Programmable Input Gain Stage** | | 36 | | | | |
| Gain (at 0dB) | | 37 | –0.5 | 0 | +0.5 | dB |
| Cumulative Gain Error (wrt attenuation at 0dB) | | 37 | –1.0 | 0 | +1.0 | dB |
| **Modulator Outputs (MOD 1, MOD 2)** | | | | | | |
| Power-up to output stable | | 41 | – | 50 | 100 | µs |
| **Modulator Attenuators** | | | | | | |
| Attenuation (at 0dB) | | 43 | –1.0 | 0 | +1.0 | dB |
| Cumulative Attenuation Error (wrt attenuation at 0dB) | | | –0.6 | 0 | +0.6 | dB |
| Output Impedance | Enabled | 42 | – | 600 | – | Ω |
| | Disabled | 42 | – | 500 | – | kΩ |
| Output current range ($AV_{DD}$ = 3.3V) | | | –125 | – | +125 | µA |
| Output voltage range | | 44 | 0.5 | – | $AV_{DD}$ –0.5 | V |
| Load resistance | | | 20 | – | – | kΩ |
| **Auxiliary Signal Inputs (Aux ADC 1 to 4)** | | | | | | |
| Source Output Impedance | | 51 | – | – | 24 | kΩ |

| AC Parameters | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **Auxiliary 10 Bit ADCs** | | | | | |
| Resolution | | – | 10 | – | Bits |
| Input Range | | 10% | – | 90% | $V_{DD}$ |
| Conversion time | | – | 21 | – | µs |
| Input impedance | | | | | |
| Resistance | | – | 10 | – | MΩ |
| Capacitance | | – | 5 | – | pF |
| Zero error (input offset to give ADC output = 0) | | – | 0 | ±10 | mV |
| Integral Non-linearity | | – | – | ±4 | LSB |
| Differential Non-linearity | 53 | – | – | ±3 | LSB |
| **Auxiliary 10 Bit DACs** | | | | | |
| Resolution | | – | 10 | – | Bits |
| Settling time (to ±0.5 LSB) | | – | 10 | – | µs |
| Output Range | | 10% | – | 90% | $V_{DD}$ |
| Output Impedance | | – | 250 | – | kΩ |
| Integral Non-linearity | | – | – | ±4 | LSB |
| Differential Non-linearity | 53 | – | – | ±1 | LSB |
| Resistive load | | 5 | – | – | kΩ |
| Noise output voltage in 30kHz bandwidth | | – | 5 | – | µVrms |
| **RF Synthesizer – Phase Locked Loops (CMX7032 only)** | | | | | |
| *Reference Clock Input* | | | | | |
| Frequency | | 5.0 | 19.2 | 40.0 | MHz |
| Level | 61 | 0.5 | – | 2.0 | Vp-p |
| Divide ratios (R) | 62 | 1 | – | 8192 | |
| *RF Synthesizer* | | | | | |
| Comparison frequency | 64 | – | – | 500 | kHz |
| Input frequency range | | 100 | – | 600 | MHz |
| Input level | | -15 | – | 0 | dBm |
| Divide ratios (N) | | 90 | – | 1048576 | |
| Phase Noise Floor | | – | –158 | – | dBc/Hz |
| Charge pump current (high) | 63 | – | ±2.5 | – | mA |
| Charge pump current (low) | 63 | – | ±625 | – | µA |
| Charge pump current - temperature variation | | – | 10% | – | per °C |
| Charge pump current - voltage variation | | – | 10% | – | per V |
| Charge pump current - sink to source match | | – | 5% | – | of ISET |
| Supply Current | | – | 3.5 | – | mA |

**Notes:**

21 Not including any current drawn from the device pins by external circuitry.
22 RF and Auxiliary circuits disabled.
25 Characteristics when driving the CLOCK/XTAL pin with an external clock source.
26 Applies when utilising $V_{BIAS}$ to provide a reference voltage to other parts of the system. When using $V_{BIAS}$ as a reference, $V_{BIAS}$ must be buffered. $V_{BIAS}$ must always be decoupled with a capacitor as shown in Figure 4.
31 Timing for an external input to the CLOCK/XTAL pin.
34 With no external components connected
35 After multiplying by gain of input circuit, with external components connected.
36 Gain applied to signal at output of buffer amplifier: DiscFB, AltFB OR MicFB
37 Design Value. Overall attenuation input to output has a tolerance of 0dB ±1.0dB
41 Power-up refers to issuing a C-BUS command to turn on an output. These limits apply only if $V_{BIAS}$ is on and stable.
42 Small signal impedance, at $AV_{DD}$ = 3.3V and Tamb = 25°C.
43 With respect to the signal at the feedback pin of the selected input port.
44 With the output driving a 20kΩ load to $AV_{DD}/2$.
51 Denotes output impedance of the driver of the auxiliary input signal, to ensure < 1 bit additional error under nominal conditions.
53 Guaranteed monotonic with no missing codes.
61 Sine wave or clipped sine wave.
62 Separate dividers provided for each PLL.
63 External ISET resistor (R31) = 0Ω (Internal ISET resistor = 9k6Ω nominally).
64 For optimum performance of the synthesizer subsystems, a common master clock should be used for the RF Synthesizers and the baseband sections. Using unsynchronised clocks is likely to result in spurious products being generated in the synthesizer outputs and in some cases difficulty may be experienced in obtaining lock in the RF Synthesizers.

### 8.1.4 Parametric Performance

For the following conditions unless otherwise specified:

External components as recommended in **Figure 2**.
Maximum load on digital outputs = 30pF.
CLK Frequency = 19.2MHz (±20ppm); Tamb = –40°C to +85°C.
$AV_{DD}$ = $DV_{DD}$ = $CPV_{DD}$ = 3.0V to 3.6V; $RFV_{DD}$ = 2.25V to 2.75V.
Reference Signal Level = 300mV pk-pk with $AV_{DD}$ = 3.3V.
Signal levels track with supply voltage, so scale accordingly.
Signal to Noise Ratio (SNR) in bit rate bandwidth.
Input stage gain = 0dB, Output stage attenuation = 0dB.

| Transmit Parameters | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **AIS (GMSK 9600bps), 25kHz channel** | | | | | |
| Bit rate accuracy | | – | – | ±50 | ppm |
| BT | | – | 0.4 | – | |
| Storage time | 1 | – | 8 | – | bits |
| Tx Buffer size | | – | – | 176 | bytes |

| Receive Parameters | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **AIS (GMSK 9600bps), 25kHz channel** | | | | | |
| Bit rate accuracy | | – | – | ±50 | ppm |
| BT | | – | 0.4 | – | |
| Storage time | 2 | – | 8 | – | bits |
| Packet error rate (PER) limit | | – | – | 20% | |
| PER with –10dB co-channel interference | 3 | – | – | 20% | |
| PER with 10dB SNR | 4 | – | – | 20% | |
| Rx Buffer Size (burst mode) | | – | – | 2 x176 | bytes |
| Rx Buffer Size (raw mode) | | – | – | 184 | bytes |
| **DSC (FSK 1200bps, 6dB/octave de-emphasis)** | | | | | |
| Bit rate accuracy | | – | – | ±50 | ppm |
| Sub-carrier | | – | 1700 | – | Hz |
| Tx mark frequency | | 1290 | 1300 | 1310 | Hz |
| Tx space frequency | | 2090 | 2100 | 2110 | Hz |
| Storage time | 2 | – | TBD | – | bits |
| Bit error rate (BER) with 10dB SNR | 4 | – | – | 1% | |

Notes:
1. Through GMSK/FSK transmit filter.
2. Through GMSK/FSK receive filters.
3. Measured at baseband to IEC 62287-1
4. Measured at baseband with simulated FM channel noise.

## 8.2 C-BUS Timing

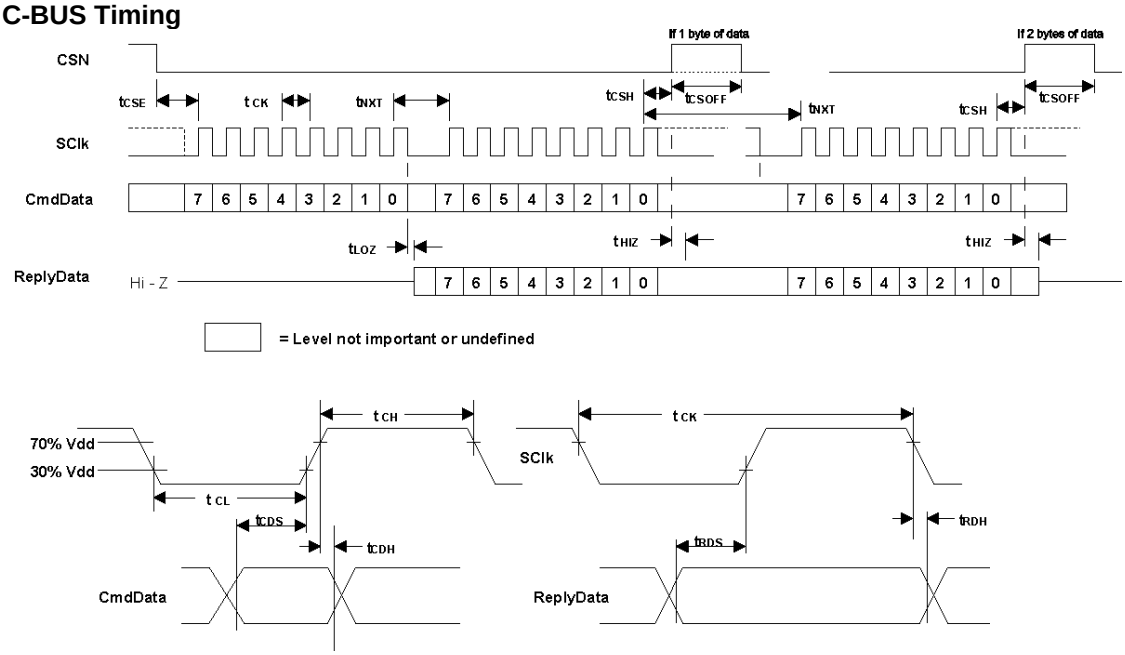



**Figure 17 C-BUS Timing**

| C-BUS Timing | | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $t_{CSE}$ | CSN Enable to SClk high time | | 100 | – | – | ns |
| $t_{CSH}$ | Last SClk high to CSN high time | | 100 | – | – | ns |
| $t_{LOZ}$ | SClk low to ReplyData Output Enable Time | | 0.0 | – | – | ns |
| $t_{HIZ}$ | CSN high to ReplyData high impedance | | – | – | 1.0 | µs |
| $t_{CSOFF}$ | CSN high time between transactions | | 1.0 | – | – | µs |
| $t_{NXT}$ | Inter-byte time | | 200 | – | – | ns |
| $t_{CK}$ | SClk cycle time | | 200 | – | – | ns |
| $t_{CH}$ | SClk high time | | 100 | – | – | ns |
| $t_{CL}$ | SClk low time | | 100 | – | – | ns |
| $t_{CDS}$ | Command Data setup time | | 75 | – | – | ns |
| $t_{CDH}$ | Command Data hold time | | 25 | – | – | ns |
| $t_{RDS}$ | Reply Data setup time | | 50 | – | – | ns |
| $t_{RDH}$ | Reply Data hold time | | 0 | – | – | ns |

**Notes:**
1. Depending on the command, 1 or 2 bytes of COMMAND DATA are transmitted to the peripheral MSB (Bit 7) first, LSB (Bit 0) last. REPLY DATA is read from the peripheral MSB (Bit 7) first, LSB (Bit 0) last.
2. Data is clocked into the peripheral on the rising SerialCLK edge.
3. Commands are acted upon at the end of each command (rising edge of CSN).
4. To allow for differing µC serial interface formats C-BUS compatible ICs are able to work with SerialCLK pulses starting and ending at either polarity.
5. Maximum 30pF load on IRQN pin and each C-BUS interface line.

These timings are for the latest version of C-BUS and allow faster transfers than the original C-BUS timing specification. The CMX7032 can be used in conjunction with devices that comply with the slower timings, subject to system throughput constraints.

### 8.3 SPI Timing

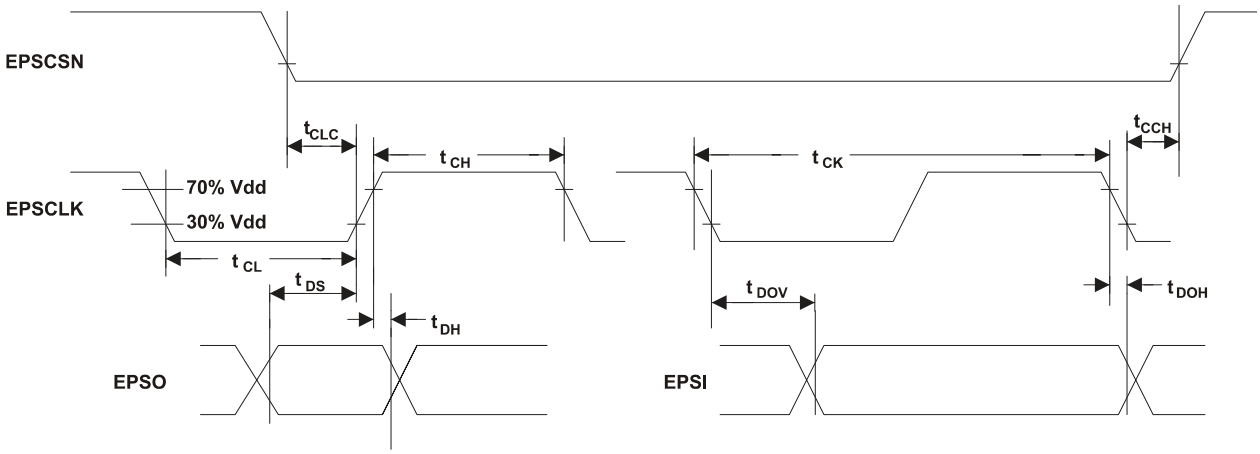
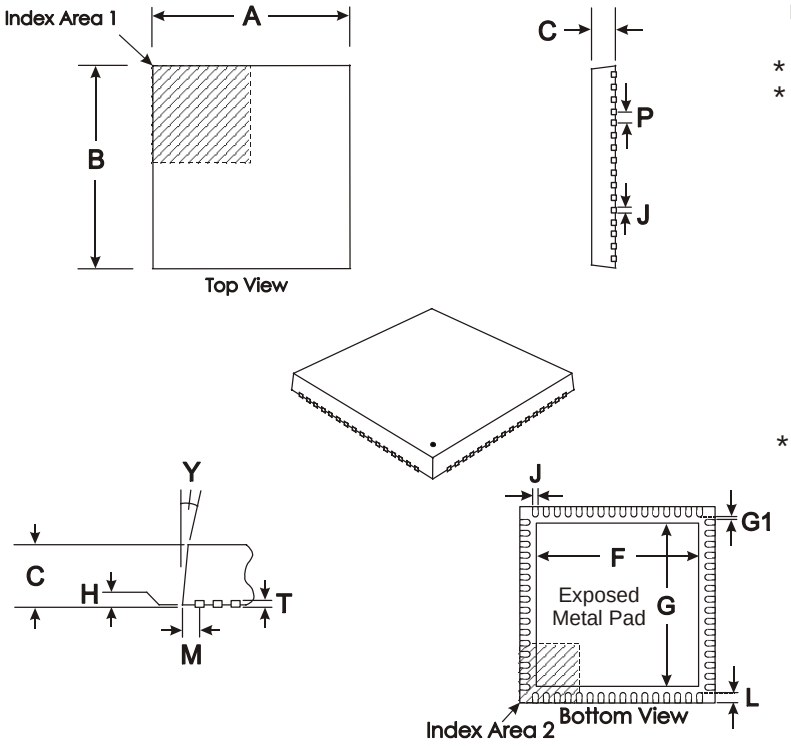



**Figure 18 SPI Interface Timing**

| Serial (SPI) Bus Interface Timing | | Notes | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $t_{CK}$ | Clock cycle time | | – | 16 | – | Xtal Clock Periods |
| $t_{CL}$ | Clock 'low' pulse width | | – | 8 | – | Xtal Clock Periods |
| $t_{CH}$ | Clock 'high' pulse width | | – | 8 | – | Xtal Clock Periods |
| $t_{DOV}$ | Out data valid time | | – | 10 | 80 | ns |
| $t_{DOH}$ | Out data hold time | | 0 | – | – | ns |
| $t_{DS}$ | In data set up time | | 20 | – | – | ns |
| $t_{DH}$ | In data hold time | | 20 | – | – | ns |
| $t_{CLC}$ | Chip select low to clock rising edge | | – | 4 | – | Xtal Clock Periods |
| $t_{CCH}$ | Clock falling edge to chip select high | | – | 2 | – | Xtal Clock Periods |

**Notes:** 1. The serial (SPI) bus clock frequency is the CMX7032 internal (Main Clock ÷ 16) frequency. When the CMX7032 is first powered up, the internal Main Clock is connected directly to the Reference Clock (the clock source present on the CLOCK pin). An EEPROM should be chosen which is compatible with these timings.
2. Maximum 30pF load on each serial bus interface line.




| | DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|---|
| * | A | | 9.00 BSC | |
| * | B | | 9.00 BSC | |
| | C | 0.80 | | 1.00 |
| | F | 7.00 | | 7.95 |
| | G | 7.00 | | 7.95 |
| | G1 | | 0.225 | |
| | H | 0.00 | | 0.05 |
| | J | 0.18 | | 0.30 |
| | L | 0.30 | | 0.50 |
| | M | | 0.75 | |
| | P | | 0.50 | |
| | T | | 0.20 | |
| | Y | 0° | | 12° |

NOTE :

* A & B are reference data and do not include mold deflash or protrusions.

All dimensions in mm
Angles are in degrees

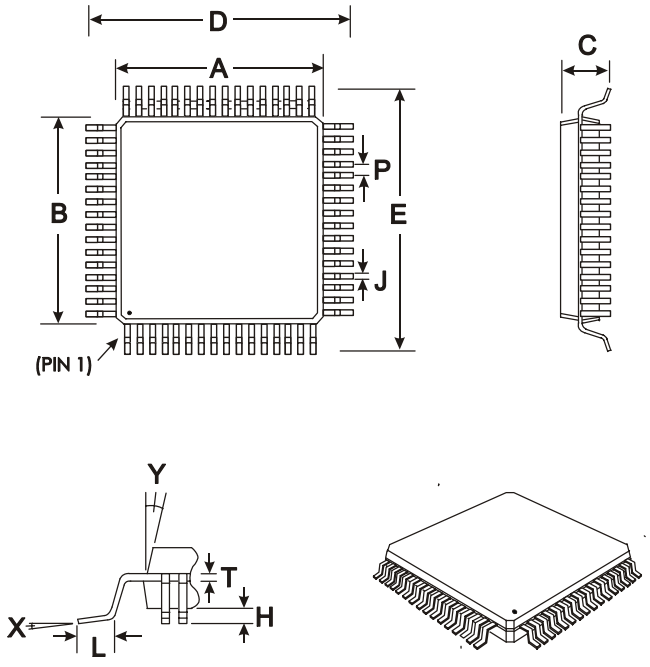
Index Area 1 Index Area 2

Index Area 1 is located directly above Index Area 2

The underside of the package has an exposed metal pad which can be soldered to the pcb to enhance the thermal conductivity and mechanical strength of the package fixing. Where advised, an electrical connection to this metal pad may also be required

**Figure 19 Mechanical outline for 64-pad VQFN package (Q1)**

**Order as CMX7032Q1**




| | DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|---|
| * | A | 9.80 | | 10.20 |
| * | B | 9.80 | | 10.20 |
| | C | 1.40 | | 1.60 |
| | D | 11.80 | | 12.20 |
| | E | 11.80 | | 12.20 |
| | H | 0.05 | | 0.15 |
| | J | 0.17 | | 0.27 |
| | L | 0.45 | | 0.75 |
| | P | | 0.50 | |
| | T | 0.09 | | 0.20 |
| | X | 0° | | 7° |
| | Y | 11° | | 13° |

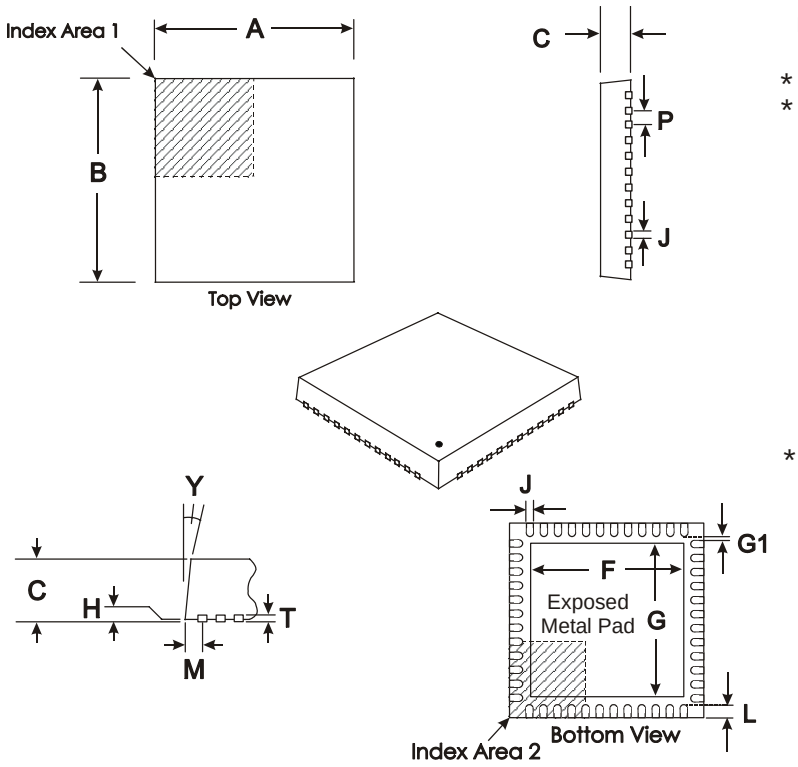
NOTE :

* A & B are reference data and do not include mold deflash or protrusions.

All dimensions in mm
Angles are in degrees
Co-Planarity of leads within 0.1mm

**Figure 20 Mechanical outline for 64-pin LQFP (leaded) package (L9)**

**Order as CMX7032L9**

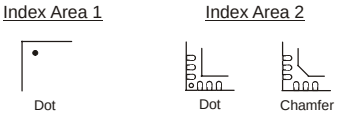



| DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|
| * A | | 7.00 BSC | |
| * B | | 7.00 BSC | |
| C | 0.80 | | 1.00 |
| F | 2.75 | | 5.25 |
| G | 2.75 | | 5.25 |
| G1 | | 0.225 | |
| H | 0.00 | | 0.05 |
| J | 0.18 | | 0.30 |
| L | 0.30 | | 0.50 |
| M | | 0.75 | |
| P | | 0.50 | |
| T | | 0.20 | |
| Y | 0° | | 12° |

NOTE :

* A & B are reference data and do not include mold deflash or protrusions.

All dimensions in mm
Angles are in degrees

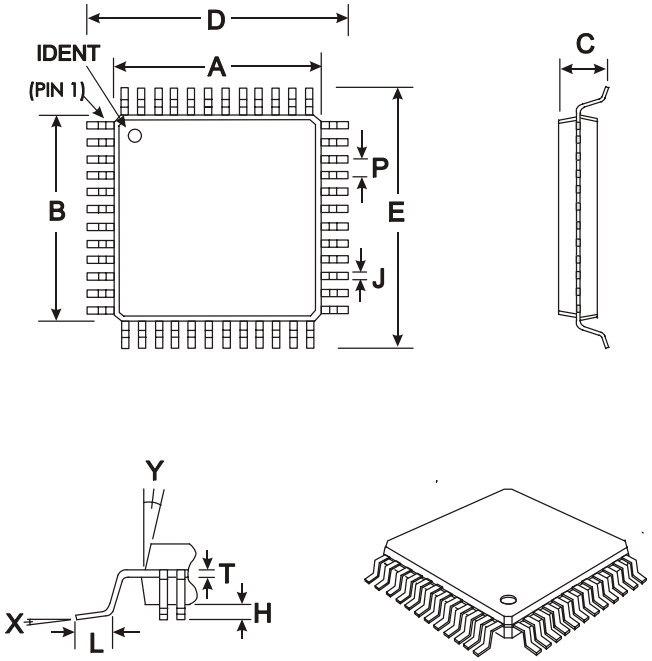



Index Area 1 is located directly above Index Area 2

The underside of the package has an exposed metal pad which can be soldered to the pcb to enhance the thermal conductivity and mechanical strength of the package fixing. Where advised, an electrical connection to this metal pad may also be required

**Figure 21 Mechanical outline for 48-pad VQFN package (Q3)**

**Order as CMX7042Q3**




| DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|
| * A | 6.91 | | 7.11 |
| * B | 6.91 | | 7.11 |
| C | 1.40 | | 1.60 |
| D | 8.74 | | 9.25 |
| E | 8.74 | | 9.25 |
| H | 0.05 | | 0.15 |
| J | 0.10 | | 0.28 |
| L | 0.35 | | 0.76 |
| P | | 0.50 | |
| T | | 0.13 | |
| X | 0° | | 7° |
| Y | 11° | | 13° |

NOTE :

* A & B are reference data and do not include mold deflash or protrusions.

All dimensions in mm
Angles are in degrees
Co-Planarity of leads within 0.1mm

**Figure 22 Mechanical outline for 48-pin LQFP (leaded) package (L4)**

**Order as CMX7042L4**

## About *FirmASIC*®

CML's proprietary *FirmASIC*® component technology reduces cost, time to market and development risk, with increased flexibility for the designer and end application. *FirmASIC*® combines Analogue, Digital, Firmware and Memory technologies in a single silicon platform that can be focused to deliver the right feature mix, performance and price for a target application family. Specific functions of a *FirmASIC*® device are determined by uploading its Function Image™ during device initialization. New Function Images™ may be later provided to supplement and enhance device functions, expanding or modifying end-product features without the need for expensive and time-consuming design changes. *FirmASIC*® devices provide significant time to market and commercial benefits over Custom ASIC, Structured ASIC, FPGA and DSP solutions. They may also be exclusively customised where security or intellectual property issues prevent the use of Application Specific Standard Products (ASSP's).

Handling precautions: This product includes input protection, however, precautions should be taken to prevent device damage from electro-static discharge. CML does not assume any responsibility for the use of any circuitry described. No IPR or circuit patent licences are implied. CML reserves the right at any time without notice to change the said circuitry and this product specification. CML has a policy of testing every product shipped using calibrated test equipment to ensure compliance with this product specification. Specific testing of all circuit parameters is not necessarily performed.

## www.cmlmicro.com

**For FAQs see:** www.cmlmicro.com/products/faqs/

**For a full data sheet listing see:** www.cmlmicro.com/products/datasheets/download.htm

**For detailed application notes:** www.cmlmicro.com/products/applications/


| CML Microcircuits (UK) Ltd<br>COMMUNICATION SEMICONDUCTORS | CML Microcircuits (USA) Inc.<br>COMMUNICATION SEMICONDUCTORS | CML Microcircuits (Singapore) Pte Ltd<br>COMMUNICATION SEMICONDUCTORS | |
|---|---|---|---|
| Oval Park, Langford,<br>Maldon, Essex,<br>CM9 6WG - England. | 4800 Bethania Station Road,<br>Winston-Salem,<br>NC 27105 - USA. | No 2 Kallang Pudding Road, #09 - 05/06 Mactech Industrial Building,<br>Singapore 349307 | No. 218, Tian Mu Road West, Tower 1, Unit 1008,<br>Shanghai Kerry Everbright City, Zhabei,<br>Shanghai 200070,<br>China. |
| Tel: +44 (0)1621 875500 | Tel: +1 336 744 5050,<br>800 638 5577 | Tel: +65 6745 0426 | Tel: +86 21 6317 4107<br>+86 21 6317 8916 |
| Fax: +44 (0)1621 875600 | Fax: +1 336 744 5054 | Fax: +65 6745 2917 | Fax: +86 21 6317 0243 |
| Sales:<br>sales@cmlmicro.com | Sales:<br>us.sales@cmlmicro.com | Sales:<br>sg.sales@cmlmicro.com | Sales:<br>cn.sales@cmlmicro.com.cn |
| Technical Support:<br>techsupport@cmlmicro.com | Technical Support:<br>us.techsupport@cmlmicro.com | Technical Support:<br>sg.techsupport@cmlmicro.com | Technical Support:<br>sg.techsupport@cmlmicro.com |